Problem Number: 0 URL: https://leetcode.com/problems/two-sum Title: 1. Two Sum Problem Description: Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order. Example 1: Input: nums = [2,7,11,15], target = 9 Output: [0,1] Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2: Input: nums = [3,2,4], target = 6 Output: [1,2]

Example 3: Input: nums = [3,3], target = 6 Output: [0,1]

Constraints:

2 <= nums.length <= 104 -109 <= nums[i] <= 109 -109 <= target <= 109 Only one valid answer exists.

Follow-up: Can you come up with an algorithm that is less than O(n2) time complexity? == Problem Number: 1 URL: https://leetcode.com/problems/palindrome-number Title: 9. Palindrome Number Problem Description: Given an integer x, return true if x is a palindrome, and false otherwise. Example 1: Input: x = 121 Output: true Explanation: 121 reads as 121 from left to right and from right to left.

Example 2: Input: x = -121 Output: false Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.

Example 3: Input: x = 10 Output: false Explanation: Reads 01 from right to left. Therefore it is not a palindrome.

Constraints:

-231 <= x <= 231 - 1

Follow up: Could you solve it without converting the integer to a string? == Problem Number: 2 URL: https://leetcode.com/problems/roman-to-integer Title: 13. Roman to Integer Problem Description: Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M. Symbol Value I 1 V 5 X 10 L 50 C 100 D 500 M 1000 For example, 2 is written as II in Roman numeral, just two ones added together. 12 is written as XII, which is simply X + II. The number 27 is written as XXVII, which is XX + V + II. Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9. X can be placed before L (50) and C (100) to make 40 and 90. C can be placed before D (500) and M (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer. Example 1: Input: s = ”III” Output: 3 Explanation: III = 3.

Example 2: Input: s = ”LVIII” Output: 58 Explanation: L = 50, V= 5, III = 3.

Example 3: Input: s = ”MCMXCIV” Output: 1994 Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

Constraints:

1 <= s.length <= 15 s contains only the characters (’I’, ’V’, ’X’, ’L’, ’C’, ’D’, ’M’). It is guaranteed that s is a valid roman numeral in the range [1, 3999].

==
Problem Number: 3 URL: https://leetcode.com/problems/longest-common-prefix Title: 14. Longest Common Prefix Problem Description: Write a function to find the longest common prefix string amongst an array of strings. If there is no common prefix, return an empty string ””. Example 1: Input: strs = [”flower”,”flow”,”flight”] Output: ”fl”

Example 2: Input: strs = [”dog”,”racecar”,”car”] Output: ”” Explanation: There is no common prefix among the input strings.

Constraints:

1 <= strs.length <= 200 0 <= strs[i].length <= 200 strs[i] consists of only lowercase English letters.

==
Problem Number: 4 URL: https://leetcode.com/problems/valid-parentheses Title: 20. Valid Parentheses Problem Description: Given a string s containing just the characters ’(’, ’)’, ’{’, ’}’, ’[’ and ’]’, determine if the input string is valid. An input string is valid if:

Open brackets must be closed by the same type of brackets. Open brackets must be closed in the correct order. Every close bracket has a corresponding open bracket of the same type.

Example 1: Input: s = ”()” Output: true

Example 2: Input: s = ”()[]{}” Output: true

Example 3: Input: s = ”(]” Output: false

Constraints:

1 <= s.length <= 104 s consists of parentheses only ’()[]{}’.

==
Problem Number: 5 URL: https://leetcode.com/problems/merge-two-sorted-lists Title: 21. Merge Two Sorted Lists Problem Description: You are given

the heads of two sorted linked lists list1 and list2. Merge the two lists into one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list.   Example 1:

Input: list1 = [1,2,4], list2 = [1,3,4] Output: [1,1,2,3,4,4]

Example 2: Input: list1 = [], list2 = [] Output: []

Example 3: Input: list1 = [], list2 = [0] Output: [0]

Constraints:

The number of nodes in both lists is in the range [0, 50]. -100 <= Node.val <= 100 Both list1 and list2 are sorted in non-decreasing order.

==
Problem Number: 6 URL: https://leetcode.com/problems/remove-duplicates-from-sorted-array Title: 26. Remove Duplicates from Sorted Array Problem Description: Given an integer array nums sorted in non-decreasing order, remove the duplicates in-place such that each unique element appears only once. The relative order of the elements should be kept the same. Then return the number of unique elements in nums. Consider the number of unique elements of nums to be k, to get accepted, you need to do the following things:

Change the array nums such that the first k elements of nums contain the unique elements in the order they were present in nums initially. The remaining elements of nums are not important as well as the size of nums. Return k.

Custom Judge: The judge will test your solution with the following code: int[] nums = [...]; // Input array int[] expectedNums = [...]; // The expected answer with correct length

int k = removeDuplicates(nums); // Calls your implementation

assert k == expectedNums.length; for (int i = 0; i < k; i++) { assert nums[i] == expectedNums[i]; }

If all assertions pass, then your solution will be accepted.   Example 1: Input: nums = [1,1,2] Output: 2, nums = [1,2,_] Explanation: Your function should return k = 2, with the first two elements of nums being 1 and 2 respectively. It does not matter what you leave beyond the returned k (hence they are underscores).

Example 2: Input: nums = [0,0,1,1,1,2,2,3,3,4] Output: 5, nums = [0,1,2,3,4,_,_,_,_,_] Explanation: Your function should return k = 5, with the first five elements of nums being 0, 1, 2, 3, and 4 respectively. It does not matter what you leave beyond the returned k (hence they are underscores).

Constraints:

1 <= nums.length <= 3 * 104 -100 <= nums[i] <= 100 nums is sorted in non-decreasing order.

==

Problem Number: 7 URL: https://leetcode.com/problems/remove-element Title: 27. Remove Element Problem Description: Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val. Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:

Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums. Return k.

Custom Judge: The judge will test your solution with the following code: int[] nums = [...]; // Input array int val = ...; // Value to remove int[] expectedNums = [...]; // The expected answer with correct length. // It is sorted with no values equaling val.

int k = removeElement(nums, val); // Calls your implementation

assert k == expectedNums.length; sort(nums, 0, k); // Sort the first k elements of nums for (int i = 0; i < actualLength; i++) { assert nums[i] == expectedNums[i]; }

If all assertions pass, then your solution will be accepted. Example 1: Input: nums = [3,2,2,3], val = 3 Output: 2, nums = [2,2,_,_] Explanation: Your function should return k = 2, with the first two elements of nums being 2. It does not matter what you leave beyond the returned k (hence they are underscores).

Example 2: Input: nums = [0,1,2,2,3,0,4,2], val = 2 Output: 5, nums = [0,1,4,0,3,_,_,_] Explanation: Your function should return k = 5, with the first five elements of nums containing 0, 0, 1, 3, and 4. Note that the five elements can be returned in any order. It does not matter what you leave beyond the returned k (hence they are underscores).

Constraints:

0 <= nums.length <= 100 0 <= nums[i] <= 50 0 <= val <= 100

==

Problem Number: 8 URL: https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string Title: 28. Find the Index of the First Occurrence in a String Problem Description: Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack. Example 1: Input: haystack = "sadbutsad", needle = "sad" Output: 0 Explanation: "sad" occurs at index 0 and 6. The first occurrence is at index 0, so we return 0.

Example 2: Input: haystack = "leetcode", needle = "leeto" Output: -1 Explanation: "leeto" did not occur in "leetcode", so we return -1.

Constraints:

1 <= haystack.length, needle.length <= 104 haystack and needle consist of only lowercase English characters.

==
Problem Number: 9 URL: https://leetcode.com/problems/search-insert-position Title: 35. Search Insert Position Problem Description: Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order. You must write an algorithm with O(log n) runtime complexity. Example 1: Input: nums = [1,3,5,6], target = 5 Output: 2

Example 2: Input: nums = [1,3,5,6], target = 2 Output: 1

Example 3: Input: nums = [1,3,5,6], target = 7 Output: 4

Constraints:

1 <= nums.length <= 104 -104 <= nums[i] <= 104 nums contains distinct values sorted in ascending order. -104 <= target <= 104

==
Problem Number: 10 URL: https://leetcode.com/problems/length-of-last-word Title: 58. Length of Last Word Problem Description: Given a string s consisting of words and spaces, return the length of the last word in the string. A word is a maximal substring consisting of non-space characters only. Example 1: Input: s = "Hello World" Output: 5 Explanation: The last word is "World" with length 5.

Example 2: Input: s = " fly me to the moon " Output: 4 Explanation: The last word is "moon" with length 4.

Example 3: Input: s = "luffy is still joyboy" Output: 6 Explanation: The last word is "joyboy" with length 6.

Constraints:

1 <= s.length <= 104 s consists of only English letters and spaces ' '. There will be at least one word in s.

==
Problem Number: 11 URL: https://leetcode.com/problems/plus-one Title: 66. Plus One Problem Description: You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's. Increment the large integer by one and return the resulting array of digits. Example 1: Input: digits = [1,2,3] Output: [1,2,4] Explanation: The array represents the integer 123. Incrementing by one gives 123 + 1 = 124. Thus, the result should be [1,2,4].

Example 2: Input: digits = [4,3,2,1] Output: [4,3,2,2] Explanation: The array represents the integer 4321. Incrementing by one gives 4321 + 1 = 4322. Thus, the result should be [4,3,2,2].

Example 3: Input: digits = [9] Output: [1,0] Explanation: The array represents the integer 9. Incrementing by one gives 9 + 1 = 10. Thus, the result should be [1,0].

Constraints:

1 <= digits.length <= 100 0 <= digits[i] <= 9 digits does not contain any leading 0's.

==
Problem Number: 12 URL: https://leetcode.com/problems/add-binary Title: 67. Add Binary Problem Description: Given two binary strings a and b, return their sum as a binary string. Example 1: Input: a = "11", b = "1" Output: "100" Example 2: Input: a = "1010", b = "1011" Output: "10101"

Constraints:

1 <= a.length, b.length <= 104 a and b consist only of '0' or '1' characters. Each string does not contain leading zeros except for the zero itself.

==
Problem Number: 13 URL: https://leetcode.com/problems/sqrtx Title: 69. Sqrt(x) Problem Description: Given a non-negative integer x, return the square root of x rounded down to the nearest integer. The returned integer should be non-negative as well. You must not use any built-in exponent function or operator.

For example, do not use pow(x, 0.5) in c++ or x ** 0.5 in python.

Example 1: Input: x = 4 Output: 2 Explanation: The square root of 4 is 2, so we return 2.

Example 2: Input: x = 8 Output: 2 Explanation: The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.

Constraints:

0 <= x <= 231 - 1

==
Problem Number: 14 URL: https://leetcode.com/problems/climbing-stairs Title: 70. Climbing Stairs Problem Description: You are climbing a staircase. It takes n steps to reach the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top? Example 1: Input: n = 2 Output: 2 Explanation: There are two ways to climb to the top. 1. 1 step + 1 step 2. 2 steps

Example 2: Input: n = 3 Output: 3 Explanation: There are three ways to climb to the top. 1. 1 step + 1 step + 1 step 2. 1 step + 2 steps 3. 2 steps + 1 step

Constraints:

1 <= n <= 45

===

Problem Number: 15 URL: https://leetcode.com/problems/remove-duplicates-from-sorted-list Title: 83. Remove Duplicates from Sorted List Problem Description: Given the head of a sorted linked list, delete all duplicates such that each element appears only once. Return the linked list sorted as well. Example 1:

Input: head = [1,1,2] Output: [1,2]

Example 2:

Input: head = [1,1,2,3,3] Output: [1,2,3]

Constraints:

The number of nodes in the list is in the range [0, 300]. -100 <= Node.val <= 100 The list is guaranteed to be sorted in ascending order.

===

Problem Number: 16 URL: https://leetcode.com/problems/merge-sorted-array Title: 88. Merge Sorted Array Problem Description: You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively. Merge nums1 and nums2 into a single array sorted in non-decreasing order. The final sorted array should not be returned by the function, but instead be stored inside the array nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n. Example 1: Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3 Output: [1,2,2,3,5,6] Explanation: The arrays we are merging are [1,2,3] and [2,5,6]. The result of the merge is [1,2,2,3,5,6] with the underlined elements coming from nums1.

Example 2: Input: nums1 = [1], m = 1, nums2 = [], n = 0 Output: [1] Explanation: The arrays we are merging are [1] and []. The result of the merge is [1].

Example 3: Input: nums1 = [0], m = 0, nums2 = [1], n = 1 Output: [1] Explanation: The arrays we are merging are [] and [1]. The result of the merge is [1]. Note that because m = 0, there are no elements in nums1. The 0 is only there to ensure the merge result can fit in nums1.

Constraints:

nums1.length == m + n nums2.length == n 0 <= m, n <= 200 1 <= m + n <= 200 -109 <= nums1[i], nums2[j] <= 109

Follow up: Can you come up with an algorithm that runs in O(m + n) time?

===

Problem Number: 17 URL: https://leetcode.com/problems/binary-tree-inorder-traversal Title: 94. Binary Tree Inorder Traversal Problem Description: Given the root of a binary tree, return the inorder traversal of its nodes' values. Example 1:

Input: root = [1,null,2,3] Output: [1,3,2]

Example 2: Input: root = [] Output: []

Example 3: Input: root = [1] Output: [1]

Constraints:

The number of nodes in the tree is in the range [0, 100]. -100 <= Node.val <= 100

Follow up: Recursive solution is trivial, could you do it iteratively? ==

Problem Number: 18 URL: https://leetcode.com/problems/same-tree Title: 100. Same Tree Problem Description: Given the roots of two binary trees p and q, write a function to check if they are the same or not. Two binary trees are considered the same if they are structurally identical, and the nodes have the same value. Example 1:

Input: p = [1,2,3], q = [1,2,3] Output: true

Example 2:

Input: p = [1,2], q = [1,null,2] Output: false

Example 3:

Input: p = [1,2,1], q = [1,1,2] Output: false

Constraints:

The number of nodes in both trees is in the range [0, 100]. -104 <= Node.val <= 104

==

Problem Number: 19 URL: https://leetcode.com/problems/symmetric-tree Title: 101. Symmetric Tree Problem Description: Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center). Example 1:

Input: root = [1,2,2,3,4,4,3] Output: true

Example 2:

Input: root = [1,2,2,null,3,null,3] Output: false

Constraints:

The number of nodes in the tree is in the range [1, 1000]. -100 <= Node.val <= 100

Follow up: Could you solve it both recursively and iteratively? ================================
Problem Number: 20 URL: https://leetcode.com/problems/maximum-depth-of-binary-tree Title: 104. Maximum Depth of Binary Tree Problem Description: Given the root of a binary tree, return its maximum depth. A binary tree's maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: 3

Example 2: Input: root = [1,null,2] Output: 2

Constraints:

The number of nodes in the tree is in the range [0, 104]. -100 <= Node.val <= 100

==
Problem Number: 21 URL: https://leetcode.com/problems/convert-sorted-array-to-binary-search-tree Title: 108. Convert Sorted Array to Binary Search Tree Problem Description: Given an integer array nums where the elements are sorted in ascending order, convert it to a height-balanced binary search tree. Example 1:

Input: nums = [-10,-3,0,5,9] Output: [0,-3,9,-10,null,5] Explanation: [0,-10,5,null,-3,null,9] is also accepted:

Example 2:

Input: nums = [1,3] Output: [3,1] Explanation: [1,null,3] and [3,1] are both height-balanced BSTs.

Constraints:

1 <= nums.length <= 104 -104 <= nums[i] <= 104 nums is sorted in a strictly increasing order.

==
Problem Number: 22 URL: https://leetcode.com/problems/balanced-binary-tree Title: 110. Balanced Binary Tree Problem Description: Given a binary tree, determine if it is height-balanced. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: true

Example 2:

Input: root = [1,2,2,3,3,null,null,4,4] Output: false

Example 3: Input: root = [] Output: true

Constraints:

The number of nodes in the tree is in the range [0, 5000]. -104 <= Node.val <= 104

==

Problem Number: 23 URL: https://leetcode.com/problems/minimum-depth-of-binary-tree Title: 111. Minimum Depth of Binary Tree Problem Description: Given a binary tree, find its minimum depth. The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node. Note: A leaf is a node with no children. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: 2

Example 2: Input: root = [2,null,3,null,4,null,5,null,6] Output: 5

Constraints:

The number of nodes in the tree is in the range [0, 105]. -1000 <= Node.val <= 1000

==

Problem Number: 24 URL: https://leetcode.com/problems/path-sum Title: 112. Path Sum Problem Description: Given the root of a binary tree and an integer targetSum, return true if the tree has a root-to-leaf path such that adding up all the values along the path equals targetSum. A leaf is a node with no children. Example 1:

Input: root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22 Output: true Explanation: The root-to-leaf path with the target sum is shown.

Example 2:

Input: root = [1,2,3], targetSum = 5 Output: false Explanation: There two root-to-leaf paths in the tree: (1 --> 2): The sum is 3. (1 --> 3): The sum is 4. There is no root-to-leaf path with sum = 5.

Example 3: Input: root = [], targetSum = 0 Output: false Explanation: Since the tree is empty, there are no root-to-leaf paths.

Constraints:

The number of nodes in the tree is in the range [0, 5000]. -1000 <= Node.val <= 1000 -1000 <= targetSum <= 1000

==

Problem Number: 25 URL: https://leetcode.com/problems/pascals-triangle Title: 118. Pascal's Triangle Problem Description: Given an integer numRows, return the first numRows of Pascal's triangle. In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:

Example 1: Input: numRows = 5 Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]] Example 2: Input: numRows = 1 Output: [[1]]

Constraints:

1 <= numRows <= 30

==
Problem Number: 26 URL: https://leetcode.com/problems/pascals-triangle-ii Title: 119. Pascal’s Triangle II Problem Description: Given an integer rowIndex, return the rowIndexth (0-indexed) row of the Pascal’s triangle. In Pascal’s triangle, each number is the sum of the two numbers directly above it as shown:

Example 1: Input: rowIndex = 3 Output: [1,3,3,1] Example 2: Input: rowIndex = 0 Output: [1] Example 3: Input: rowIndex = 1 Output: [1,1]

Constraints:

0 <= rowIndex <= 33

Follow up: Could you optimize your algorithm to use only O(rowIndex) extra space?

==
Problem Number: 27 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock Title: 121. Best Time to Buy and Sell Stock Problem Description: You are given an array prices where prices[i] is the price of a given stock on the ith day. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0. Example 1: Input: prices = [7,1,5,3,6,4] Output: 5 Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5. Note that buying on day 2 and selling on day 1 is not allowed because you must buy before you sell.

Example 2: Input: prices = [7,6,4,3,1] Output: 0 Explanation: In this case, no transactions are done and the max profit = 0.

Constraints:

1 <= prices.length <= 105 0 <= prices[i] <= 104

==
Problem Number: 28 URL: https://leetcode.com/problems/valid-palindrome Title: 125. Valid Palindrome Problem Description: A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers. Given a string s, return true if it is a palindrome, or false otherwise. Example 1: Input: s = ”A man, a plan, a canal: Panama” Output: true Explanation: ”amanaplanacanalpanama” is a palindrome.

Example 2: Input: s = ”race a car” Output: false Explanation: ”raceacar” is not a palindrome.

Example 3: Input: s = ” ” Output: true Explanation: s is an empty string ”” after removing non-alphanumeric characters. Since an empty string reads the same forward and backward, it is a palindrome.

Constraints:

1 <= s.length <= 2 * 105 s consists only of printable ASCII characters.

==

Problem Number: 29 URL: https://leetcode.com/problems/single-number Title: 136. Single Number Problem Description: Given a non-empty array of integers nums, every element appears twice except for one. Find that single one. You must implement a solution with a linear runtime complexity and use only constant extra space. Example 1: Input: nums = [2,2,1] Output: 1 Example 2: Input: nums = [4,1,2,1,2] Output: 4 Example 3: Input: nums = [1] Output: 1

Constraints:

1 <= nums.length <= 3 * 104 -3 * 104 <= nums[i] <= 3 * 104 Each element in the array appears twice except for one element which appears only once.

==

Problem Number: 30 URL: https://leetcode.com/problems/linked-list-cycle Title: 141. Linked List Cycle Problem Description: Given head, the head of a linked list, determine if the linked list has a cycle in it. There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to. Note that pos is not passed as a parameter. Return true if there is a cycle in the linked list. Otherwise, return false. Example 1:

Input: head = [3,2,0,-4], pos = 1 Output: true Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).

Example 2:

Input: head = [1,2], pos = 0 Output: true Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.

Example 3:

Input: head = [1], pos = -1 Output: false Explanation: There is no cycle in the linked list.

Constraints:

The number of the nodes in the list is in the range [0, 104]. -105 <= Node.val <= 105 pos is -1 or a valid index in the linked-list.

Follow up: Can you solve it using O(1) (i.e. constant) memory?

==

Problem Number: 31 URL: https://leetcode.com/problems/binary-tree-preorder-traversal Title: 144. Binary Tree Preorder Traversal Problem

Description: Given the root of a binary tree, return the preorder traversal of its nodes' values. Example 1:

Input: root = [1,null,2,3] Output: [1,2,3]

Example 2: Input: root = [] Output: []

Example 3: Input: root = [1] Output: [1]

Constraints:

The number of nodes in the tree is in the range [0, 100]. -100 <= Node.val <= 100

Follow up: Recursive solution is trivial, could you do it iteratively?

==

Problem Number: 32 URL: https://leetcode.com/problems/binary-tree-postorder-traversal Title: 145. Binary Tree Postorder Traversal Problem Description: Given the root of a binary tree, return the postorder traversal of its nodes' values. Example 1:

Input: root = [1,null,2,3] Output: [3,2,1]

Example 2: Input: root = [] Output: []

Example 3: Input: root = [1] Output: [1]

Constraints:

The number of the nodes in the tree is in the range [0, 100]. -100 <= Node.val <= 100

Follow up: Recursive solution is trivial, could you do it iteratively?

==

Problem Number: 33 URL: https://leetcode.com/problems/intersection-of-two-linked-lists Title: 160. Intersection of Two Linked Lists Problem Description: Given the heads of two singly linked-lists headA and headB, return the node at which the two lists intersect. If the two linked lists have no intersection at all, return null. For example, the following two linked lists begin to intersect at node c1:

The test cases are generated such that there are no cycles anywhere in the entire linked structure. Note that the linked lists must retain their original structure after the function returns. Custom Judge: The inputs to the judge are given as follows (your program is not given these inputs):

intersectVal - The value of the node where the intersection occurs. This is 0 if there is no intersected node. listA - The first linked list. listB - The second linked list. skipA - The number of nodes to skip ahead in listA (starting from the head) to get to the intersected node. skipB - The number of nodes to skip ahead in listB (starting from the head) to get to the intersected node.

The judge will then create the linked structure based on these inputs and pass the two heads, headA and headB to your program. If you correctly return the intersected node, then your solution will be accepted. Example 1:

Input: intersectVal = 8, listA = [4,1,8,4,5], listB = [5,6,1,8,4,5], skipA = 2, skipB = 3 Output: Intersected at '8' Explanation: The intersected node's value is 8 (note that this must not be 0 if the two lists intersect). From the head of A, it reads as [4,1,8,4,5]. From the head of B, it reads as [5,6,1,8,4,5]. There are 2 nodes before the intersected node in A; There are 3 nodes before the intersected node in B. - Note that the intersected node's value is not 1 because the nodes with value 1 in A and B (2nd node in A and 3rd node in B) are different node references. In other words, they point to two different locations in memory, while the nodes with value 8 in A and B (3rd node in A and 4th node in B) point to the same location in memory.

Example 2:

Input: intersectVal = 2, listA = [1,9,1,2,4], listB = [3,2,4], skipA = 3, skipB = 1 Output: Intersected at '2' Explanation: The intersected node's value is 2 (note that this must not be 0 if the two lists intersect). From the head of A, it reads as [1,9,1,2,4]. From the head of B, it reads as [3,2,4]. There are 3 nodes before the intersected node in A; There are 1 node before the intersected node in B.

Example 3:

Input: intersectVal = 0, listA = [2,6,4], listB = [1,5], skipA = 3, skipB = 2 Output: No intersection Explanation: From the head of A, it reads as [2,6,4]. From the head of B, it reads as [1,5]. Since the two lists do not intersect, intersectVal must be 0, while skipA and skipB can be arbitrary values. Explanation: The two lists do not intersect, so return null.

Constraints:

The number of nodes of listA is in the m. The number of nodes of listB is in the n. 1 <= m, n <= 3 * 104 1 <= Node.val <= 105 0 <= skipA < m 0 <= skipB < n intersectVal is 0 if listA and listB do not intersect. intersectVal == listA[skipA] == listB[skipB] if listA and listB intersect.

Follow up: Could you write a solution that runs in O(m + n) time and use only O(1) memory? == Problem Number: 34 URL: https://leetcode.com/problems/excel-sheet-column-title Title: 168. Excel Sheet Column Title Problem Description: Given an integer columnNumber, return its corresponding column title as it appears in an Excel sheet. For example: A -> 1 B -> 2 C -> 3 ... Z -> 26 AA -> 27 AB -> 28 ...

Example 1: Input: columnNumber = 1 Output: "A"

Example 2: Input: columnNumber = 28 Output: "AB"

Example 3: Input: columnNumber = 701 Output: "ZY"

Constraints:

1 <= columnNumber <= 231 - 1

==
Problem Number: 35 URL: https://leetcode.com/problems/majority-element Title: 169. Majority Element Problem Description: Given an array nums of size n, return the majority element. The majority element is the element that appears more than n / 2 times. You may assume that the majority element always exists in the array. Example 1: Input: nums = [3,2,3] Output: 3 Example 2: Input: nums = [2,2,1,1,1,2,2] Output: 2

Constraints:

n == nums.length 1 <= n <= 5 * 104 -109 <= nums[i] <= 109

Follow-up: Could you solve the problem in linear time and in O(1) space?
==
Problem Number: 36 URL: https://leetcode.com/problems/excel-sheet-column-number Title: 171. Excel Sheet Column Number Problem Description: Given a string columnTitle that represents the column title as appears in an Excel sheet, return its corresponding column number. For example: A -> 1 B -> 2 C -> 3 ... Z -> 26 AA -> 27 AB -> 28 ...

Example 1: Input: columnTitle = "A" Output: 1

Example 2: Input: columnTitle = "AB" Output: 28

Example 3: Input: columnTitle = "ZY" Output: 701

Constraints:

1 <= columnTitle.length <= 7 columnTitle consists only of uppercase English letters. columnTitle is in the range ["A", "FXSHRXW"].

==
Problem Number: 37 URL: https://leetcode.com/problems/reverse-bits Title: 190. Reverse Bits Problem Description: Reverse bits of a given 32 bits unsigned integer. Note:

Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned. In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in Example 2 above, the input represents the signed integer -3 and the output represents the signed integer -1073741825.

Example 1: Input: n = 00000010100101000001111010011100 Output: 964176192 (00111001011110000010100101000000) Explanation: The input binary string 00000010100101000001111010011100 represents the unsigned integer 43261596, so return 964176192 which its binary representation is 00111001011110000010100101000000.

Example 2: Input: n = 11111111111111111111111111111101 Output: 3221225471 (10111111111111111111111111111111) Explanation: The input binary string 11111111111111111111111111111101 represents the unsigned integer 4294967293, so return 3221225471 which its binary representation is 10111111111111111111111111111111.

Constraints:

The input must be a binary string of length 32

Follow up: If this function is called many times, how would you optimize it?
==
Problem Number: 38 URL: https://leetcode.com/problems/number-of-1-bits Title: 191. Number of 1 Bits Problem Description: Write a function that takes the binary representation of an unsigned integer and returns the number of '1' bits it has (also known as the Hamming weight). Note:

Note that in some languages, such as Java, there is no unsigned integer type. In this case, the input will be given as a signed integer type. It should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned. In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in Example 3, the input represents the signed integer. -3.

Example 1: Input: n = 00000000000000000000000000001011 Output: 3 Explanation: The input binary string 00000000000000000000000000001011 has a total of three '1' bits.

Example 2: Input: n = 00000000000000000000000010000000 Output: 1 Explanation: The input binary string 00000000000000000000000010000000 has a total of one '1' bit.

Example 3: Input: n = 11111111111111111111111111111101 Output: 31 Explanation: The input binary string 11111111111111111111111111111101 has a total of thirty one '1' bits.

Constraints:

The input must be a binary string of length 32.

Follow up: If this function is called many times, how would you optimize it?
==
Problem Number: 39 URL: https://leetcode.com/problems/happy-number Title: 202. Happy Number Problem Description: Write an algorithm to determine if a number n is happy. A happy number is a number defined by the following process:

Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy.

Return true if n is a happy number, and false if not.   Example 1: Input: n = 19 Output: true Explanation: 12 + 92 = 82 82 + 22 = 68 62 + 82 = 100 12 + 02 + 02 = 1

Example 2: Input: n = 2 Output: false

Constraints:

1 <= n <= 231 - 1

===
Problem Number: 40 URL: https://leetcode.com/problems/remove-linked-list-elements Title: 203. Remove Linked List Elements Problem Description: Given the head of a linked list and an integer val, remove all the nodes of the linked list that has Node.val == val, and return the new head.   Example 1:

Input: head = [1,2,6,3,4,5,6], val = 6 Output: [1,2,3,4,5]

Example 2: Input: head = [], val = 1 Output: []

Example 3: Input: head = [7,7,7,7], val = 7 Output: []

Constraints:

The number of nodes in the list is in the range [0, 104]. 1 <= Node.val <= 50 0 <= val <= 50

===
Problem Number: 41 URL: https://leetcode.com/problems/isomorphic-strings Title: 205. Isomorphic Strings Problem Description: Given two strings s and t, determine if they are isomorphic. Two strings s and t are isomorphic if the characters in s can be replaced to get t. All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.   Example 1: Input: s = "egg", t = "add" Output: true Example 2: Input: s = "foo", t = "bar" Output: false Example 3: Input: s = "paper", t = "title" Output: true

Constraints:

1 <= s.length <= 5 * 104 t.length == s.length s and t consist of any valid ascii character.

===
Problem Number: 42 URL: https://leetcode.com/problems/reverse-linked-list Title: 206. Reverse Linked List Problem Description: Given the head of a singly linked list, reverse the list, and return the reversed list.   Example 1:

Input: head = [1,2,3,4,5] Output: [5,4,3,2,1]

Example 2:

Input: head = [1,2] Output: [2,1]

Example 3: Input: head = [] Output: []

Constraints:

The number of nodes in the list is the range [0, 5000]. -5000 <= Node.val <= 5000

Follow up: A linked list can be reversed either iteratively or recursively. Could you implement both?

==
Problem Number: 43 URL: https://leetcode.com/problems/contains-duplicate Title: 217. Contains Duplicate Problem Description: Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct. Example 1: Input: nums = [1,2,3,1] Output: true Example 2: Input: nums = [1,2,3,4] Output: false Example 3: Input: nums = [1,1,1,3,3,4,3,2,4,2] Output: true

Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109

==
Problem Number: 44 URL: https://leetcode.com/problems/contains-duplicate-ii Title: 219. Contains Duplicate II Problem Description: Given an integer array nums and an integer k, return true if there are two distinct indices i and j in the array such that nums[i] == nums[j] and abs(i - j) <= k. Example 1: Input: nums = [1,2,3,1], k = 3 Output: true

Example 2: Input: nums = [1,0,1,1], k = 1 Output: true

Example 3: Input: nums = [1,2,3,1,2,3], k = 2 Output: false

Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109 0 <= k <= 105

==
Problem Number: 45 URL: https://leetcode.com/problems/count-complete-tree-nodes Title: 222. Count Complete Tree Nodes Problem Description: Given the root of a complete binary tree, return the number of the nodes in the tree. According to Wikipedia, every level, except possibly the last, is completely filled in a complete binary tree, and all nodes in the last level are as far left as possible. It can have between 1 and 2h nodes inclusive at the last level h. Design an algorithm that runs in less than O(n) time complexity. Example 1:

Input: root = [1,2,3,4,5,6] Output: 6

Example 2: Input: root = [] Output: 0

Example 3: Input: root = [1] Output: 1

Constraints:

The number of nodes in the tree is in the range [0, 5 * 104]. 0 <= Node.val <= 5 * 104 The tree is guaranteed to be complete.

==
Problem Number: 46 URL: https://leetcode.com/problems/implement-stack-using-queues Title: 225. Implement Stack using Queues Problem Description: Implement a last-in-first-out (LIFO) stack using only two queues. The implemented stack should support all the functions of a normal stack (push, top, pop, and empty). Implement the MyStack class:

void push(int x) Pushes element x to the top of the stack. int pop() Removes the element on the top of the stack and returns it. int top() Returns the element on the top of the stack. boolean empty() Returns true if the stack is empty, false otherwise.

Notes:

You must use only standard operations of a queue, which means that only push to back, peek/pop from front, size and is empty operations are valid. Depending on your language, the queue may not be supported natively. You may simulate a queue using a list or deque (double-ended queue) as long as you use only a queue's standard operations.

Example 1: Input ["MyStack", "push", "push", "top", "pop", "empty"] [[], [1], [2], [], [], []] Output [null, null, null, 2, 2, false]

Explanation MyStack myStack = new MyStack(); myStack.push(1); myStack.push(2); myStack.top(); // return 2 myStack.pop(); // return 2 myStack.empty(); // return False

Constraints:

1 <= x <= 9 At most 100 calls will be made to push, pop, top, and empty. All the calls to pop and top are valid.

Follow-up: Can you implement the stack using only one queue?

==
Problem Number: 47 URL: https://leetcode.com/problems/invert-binary-tree Title: 226. Invert Binary Tree Problem Description: Given the root of a binary tree, invert the tree, and return its root. Example 1:

Input: root = [4,2,7,1,3,6,9] Output: [4,7,2,9,6,3,1]

Example 2:

Input: root = [2,1,3] Output: [2,3,1]

Example 3: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 100]. -100 <= Node.val <= 100

==
Problem Number: 48 URL: https://leetcode.com/problems/summary-ranges Title: 228. Summary Ranges Problem Description: You are given a sorted unique integer array nums. A range [a,b] is the set of all integers from a to b (inclusive). Return the smallest sorted list of ranges that cover all the numbers in the array exactly. That is, each element of nums is covered by exactly one of the ranges, and there is no integer x such that x is in one of the ranges but not in nums. Each range [a,b] in the list should be output as:

"a->b" if a != b "a" if a == b

Example 1: Input: nums = [0,1,2,4,5,7] Output: ["0->2","4->5","7"] Explanation: The ranges are: [0,2] --> "0->2" [4,5] --> "4->5" [7,7] --> "7"

Example 2: Input: nums = [0,2,3,4,6,8,9] Output: ["0","2->4","6","8->9"] Explanation: The ranges are: [0,0] --> "0" [2,4] --> "2->4" [6,6] --> "6" [8,9] --> "8->9"

Constraints:

0 <= nums.length <= 20 -231 <= nums[i] <= 231 - 1 All the values of nums are unique. nums is sorted in ascending order.

==
Problem Number: 49 URL: https://leetcode.com/problems/power-of-two Title: 231. Power of Two Problem Description: Given an integer n, return true if it is a power of two. Otherwise, return false. An integer n is a power of two, if there exists an integer x such that n == 2x. Example 1: Input: n = 1 Output: true Explanation: 20 = 1

Example 2: Input: n = 16 Output: true Explanation: 24 = 16

Example 3: Input: n = 3 Output: false

Constraints:

-231 <= n <= 231 - 1

Follow up: Could you solve it without loops/recursion? ==================================
Problem Number: 50 URL: https://leetcode.com/problems/implement-queue-using-stacks Title: 232. Implement Queue using Stacks Problem Description: Implement a first in first out (FIFO) queue using only two stacks. The implemented queue should support all the functions of a normal queue (push, peek, pop, and empty). Implement the MyQueue class:

void push(int x) Pushes element x to the back of the queue. int pop() Removes the element from the front of the queue and returns it. int peek() Returns the element at the front of the queue. boolean empty() Returns true if the queue is empty, false otherwise.

Notes:

You must use only standard operations of a stack, which means only push to top, peek/pop from top, size, and is empty operations are valid. Depending on your language, the stack may not be supported natively. You may simulate a stack using a list or deque (double-ended queue) as long as you use only a stack's standard operations.

Example 1: Input ["MyQueue", "push", "push", "peek", "pop", "empty"] [[], [1], [2], [], [], []] Output [null, null, null, 1, 1, false]

Explanation MyQueue myQueue = new MyQueue(); myQueue.push(1); // queue is: [1] myQueue.push(2); // queue is: [1, 2] (leftmost is front of the queue) myQueue.peek(); // return 1 myQueue.pop(); // return 1, queue is [2] myQueue.empty(); // return false

Constraints:

1 <= x <= 9 At most 100 calls will be made to push, pop, peek, and empty. All the calls to pop and peek are valid.

Follow-up: Can you implement the queue such that each operation is amortized O(1) time complexity? In other words, performing n operations will take overall O(n) time even if one of those operations may take longer.

==
Problem Number: 51 URL: https://leetcode.com/problems/palindrome-linked-list Title: 234. Palindrome Linked List Problem Description: Given the head of a singly linked list, return true if it is a palindrome or false otherwise. Example 1:

Input: head = [1,2,2,1] Output: true

Example 2:

Input: head = [1,2] Output: false

Constraints:

The number of nodes in the list is in the range [1, 105]. 0 <= Node.val <= 9

Follow up: Could you do it in O(n) time and O(1) space? ==========================
Problem Number: 52 URL: https://leetcode.com/problems/valid-anagram Title: 242. Valid Anagram Problem Description: Given two strings s and t, return true if t is an anagram of s, and false otherwise. An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once. Example 1: Input: s =

”anagram”, t = ”nagaram” Output: true Example 2: Input: s = ”rat”, t = ”car” Output: false

Constraints:

1 <= s.length, t.length <= 5 * 104 s and t consist of lowercase English letters.

Follow up: What if the inputs contain Unicode characters? How would you adapt your solution to such a case?

==

Problem Number: 53 URL: https://leetcode.com/problems/binary-tree-paths Title: 257. Binary Tree Paths Problem Description: Given the root of a binary tree, return all root-to-leaf paths in any order. A leaf is a node with no children. Example 1:

Input: root = [1,2,3,null,5] Output: [”1->2->5”,”1->3”]

Example 2: Input: root = [1] Output: [”1”]

Constraints:

The number of nodes in the tree is in the range [1, 100]. -100 <= Node.val <= 100

==

Problem Number: 54 URL: https://leetcode.com/problems/add-digits Title: 258. Add Digits Problem Description: Given an integer num, repeatedly add all its digits until the result has only one digit, and return it. Example 1: Input: num = 38 Output: 2 Explanation: The process is 38 --> 3 + 8 --> 11 11 --> 1 + 1 --> 2 Since 2 has only one digit, return it.

Example 2: Input: num = 0 Output: 0

Constraints:

0 <= num <= 231 - 1

Follow up: Could you do it without any loop/recursion in O(1) runtime?

==

Problem Number: 55 URL: https://leetcode.com/problems/ugly-number Title: 263. Ugly Number Problem Description: An ugly number is a positive integer whose prime factors are limited to 2, 3, and 5. Given an integer n, return true if n is an ugly number. Example 1: Input: n = 6 Output: true Explanation: $6 = 2 \times 3$

Example 2: Input: n = 1 Output: true Explanation: 1 has no prime factors, therefore all of its prime factors are limited to 2, 3, and 5.

Example 3: Input: n = 14 Output: false Explanation: 14 is not ugly since it includes the prime factor 7.

Constraints:

-231 <= n <= 231 - 1

==

Problem Number: 56 URL: https://leetcode.com/problems/missing-number Title: 268. Missing Number Problem Description: Given an array nums containing n distinct numbers in the range [0, n], return the only number in the range that is missing from the array. Example 1: Input: nums = [3,0,1] Output: 2 Explanation: n = 3 since there are 3 numbers, so all numbers are in the range [0,3]. 2 is the missing number in the range since it does not appear in nums.

Example 2: Input: nums = [0,1] Output: 2 Explanation: n = 2 since there are 2 numbers, so all numbers are in the range [0,2]. 2 is the missing number in the range since it does not appear in nums.

Example 3: Input: nums = [9,6,4,2,3,5,7,0,1] Output: 8 Explanation: n = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.

Constraints:

n == nums.length 1 <= n <= 104 0 <= nums[i] <= n All the numbers of nums are unique.

Follow up: Could you implement a solution using only O(1) extra space complexity and O(n) runtime complexity?

==

Problem Number: 57 URL: https://leetcode.com/problems/first-bad-version Title: 278. First Bad Version Problem Description: You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad. Suppose you have n versions [1, 2, ..., n] and you want to find out the first bad one, which causes all the following ones to be bad. You are given an API bool isBadVersion(version) which returns whether version is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API. Example 1: Input: n = 5, bad = 4 Output: 4 Explanation: call isBadVersion(3) -> false call isBadVersion(5) -> true call isBadVersion(4) -> true Then 4 is the first bad version.

Example 2: Input: n = 1, bad = 1 Output: 1

Constraints:

1 <= bad <= n <= 231 - 1

==

Problem Number: 58 URL: https://leetcode.com/problems/move-zeroes Title: 283. Move Zeroes Problem Description: Given an integer array nums, move all 0's to the end of it while maintaining the relative order of the non-zero elements. Note that you must do this in-place without making a copy of the array. Example 1: Input: nums = [0,1,0,3,12] Output: [1,3,12,0,0] Example 2: Input: nums = [0] Output: [0]

Constraints:

1 <= nums.length <= 104 -231 <= nums[i] <= 231 - 1

Follow up: Could you minimize the total number of operations done? ==

Problem Number: 59 URL: https://leetcode.com/problems/word-pattern Title: 290. Word Pattern Problem Description: Given a pattern and a string s, find if s follows the same pattern. Here follow means a full match, such that there is a bijection between a letter in pattern and a non-empty word in s. Example 1: Input: pattern = "abba", s = "dog cat cat dog" Output: true

Example 2: Input: pattern = "abba", s = "dog cat cat fish" Output: false

Example 3: Input: pattern = "aaaa", s = "dog cat cat dog" Output: false

Constraints:

1 <= pattern.length <= 300 pattern contains only lower-case English letters. 1 <= s.length <= 3000 s contains only lowercase English letters and spaces ' '. s does not contain any leading or trailing spaces. All the words in s are separated by a single space.

==

Problem Number: 60 URL: https://leetcode.com/problems/nim-game Title: 292. Nim Game Problem Description: You are playing the following Nim Game with your friend:

Initially, there is a heap of stones on the table. You and your friend will alternate taking turns, and you go first. On each turn, the person whose turn it is will remove 1 to 3 stones from the heap. The one who removes the last stone is the winner.

Given n, the number of stones in the heap, return true if you can win the game assuming both you and your friend play optimally, otherwise return false. Example 1: Input: n = 4 Output: false Explanation: These are the possible outcomes: 1. You remove 1 stone. Your friend removes 3 stones, including the last stone. Your friend wins. 2. You remove 2 stones. Your friend removes 2 stones, including the last stone. Your friend wins. 3. You remove 3 stones. Your friend removes the last stone. Your friend wins. In all outcomes, your friend wins.

Example 2: Input: n = 1 Output: true

Example 3: Input: n = 2 Output: true

Constraints:

1 <= n <= 231 - 1

==

Problem Number: 61 URL: https://leetcode.com/problems/range-sum-query-immutable Title: 303. Range Sum Query - Immutable Problem Description: Given an integer array nums, handle multiple queries of the following type:

Calculate the sum of the elements of nums between indices left and right inclusive where left <= right.

Implement the NumArray class:

NumArray(int[] nums) Initializes the object with the integer array nums. int sumRange(int left, int right) Returns the sum of the elements of nums between indices left and right inclusive (i.e. nums[left] + nums[left + 1] + ... + nums[right]).

Example 1: Input ["NumArray", "sumRange", "sumRange", "sumRange"] [[[-2, 0, 3, -5, 2, -1]], [0, 2], [2, 5], [0, 5]] Output [null, 1, -1, -3]

Explanation NumArray numArray = new NumArray([-2, 0, 3, -5, 2, -1]); numArray.sumRange(0, 2); // return (-2) + 0 + 3 = 1 numArray.sumRange(2, 5); // return 3 + (-5) + 2 + (-1) = -1 numArray.sumRange(0, 5); // return (-2) + 0 + 3 + (-5) + 2 + (-1) = -3

Constraints:

1 <= nums.length <= 104 -105 <= nums[i] <= 105 0 <= left <= right < nums.length At most 104 calls will be made to sumRange.

==

Problem Number: 62 URL: https://leetcode.com/problems/power-of-three Title: 326. Power of Three Problem Description: Given an integer n, return true if it is a power of three. Otherwise, return false. An integer n is a power of three, if there exists an integer x such that n == 3x. Example 1: Input: n = 27 Output: true Explanation: 27 = 33

Example 2: Input: n = 0 Output: false Explanation: There is no x where 3x = 0.

Example 3: Input: n = -1 Output: false Explanation: There is no x where 3x = (-1).

Constraints:

-231 <= n <= 231 - 1

Follow up: Could you solve it without loops/recursion? =================================
Problem Number: 63 URL: https://leetcode.com/problems/counting-bits Title: 338. Counting Bits Problem Description: Given an integer n, return an array ans of length n + 1 such that for each i (0 <= i <= n), ans[i] is the number of 1's in the binary representation of i. Example 1: Input: n = 2 Output: [0,1,1] Explanation: 0 --> 0 1 --> 1 2 --> 10

Example 2: Input: n = 5 Output: [0,1,1,2,1,2] Explanation: 0 --> 0 1 --> 1 2 --> 10 3 --> 11 4 --> 100 5 --> 101

Constraints:

0 <= n <= 105

Follow up:

It is very easy to come up with a solution with a runtime of O(n log n). Can you do it in linear time O(n) and possibly in a single pass? Can you do it without using any built-in function (i.e., like ___builtin_popcount in C++)?

== Problem Number: 64 URL: https://leetcode.com/problems/power-of-four Title: 342. Power of Four Problem Description: Given an integer n, return true if it is a power of four. Otherwise, return false. An integer n is a power of four, if there exists an integer x such that n == 4x. Example 1: Input: n = 16 Output: true Example 2: Input: n = 5 Output: false Example 3: Input: n = 1 Output: true

Constraints:

-231 <= n <= 231 - 1

Follow up: Could you solve it without loops/recursion? ==================================== Problem Number: 65 URL: https://leetcode.com/problems/reverse-string Title: 344. Reverse String Problem Description: Write a function that reverses a string. The input string is given as an array of characters s. You must do this by modifying the input array in-place with O(1) extra memory. Example 1: Input: s = ["h","e","l","l","o"] Output: ["o","l","l","e","h"] Example 2: Input: s = ["H","a","n","n","a","h"] Output: ["h","a","n","n","a","H"]

Constraints:

1 <= s.length <= 105 s[i] is a printable ascii character.

== Problem Number: 66 URL: https://leetcode.com/problems/reverse-vowels-of-a-string Title: 345. Reverse Vowels of a String Problem Description: Given a string s, reverse only all the vowels in the string and return it. The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in both lower and upper cases, more than once. Example 1: Input: s = "hello" Output: "holle" Example 2: Input: s = "leetcode" Output: "leotcede"

Constraints:

1 <= s.length <= 3 * 105 s consist of printable ASCII characters.

== Problem Number: 67 URL: https://leetcode.com/problems/intersection-of-two-arrays Title: 349. Intersection of Two Arrays Problem Description: Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must be unique and you may return the result in any order. Example 1: Input: nums1 = [1,2,2,1], nums2 = [2,2] Output: [2]

Example 2: Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4] Output: [9,4] Explanation: [4,9] is also accepted.

Constraints:

1 <= nums1.length, nums2.length <= 1000 0 <= nums1[i], nums2[i] <= 1000

==

Problem Number: 68 URL: https://leetcode.com/problems/intersection-of-two-arrays-ii Title: 350. Intersection of Two Arrays II Problem Description: Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order. Example 1: Input: nums1 = [1,2,2,1], nums2 = [2,2] Output: [2,2]

Example 2: Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4] Output: [4,9] Explanation: [9,4] is also accepted.

Constraints:

1 <= nums1.length, nums2.length <= 1000 0 <= nums1[i], nums2[i] <= 1000

Follow up:

What if the given array is already sorted? How would you optimize your algorithm? What if nums1's size is small compared to nums2's size? Which algorithm is better? What if elements of nums2 are stored on disk, and the memory is limited such that you cannot load all elements into the memory at once?

==

Problem Number: 69 URL: https://leetcode.com/problems/valid-perfect-square Title: 367. Valid Perfect Square Problem Description: Given a positive integer num, return true if num is a perfect square or false otherwise. A perfect square is an integer that is the square of an integer. In other words, it is the product of some integer with itself. You must not use any built-in library function, such as sqrt. Example 1: Input: num = 16 Output: true Explanation: We return true because 4 * 4 = 16 and 4 is an integer.

Example 2: Input: num = 14 Output: false Explanation: We return false because 3.742 * 3.742 = 14 and 3.742 is not an integer.

Constraints:

1 <= num <= 231 - 1

==

Problem Number: 70 URL: https://leetcode.com/problems/guess-number-higher-or-lower Title: 374. Guess Number Higher or Lower Problem Description: We are playing the Guess Game. The game is as follows: I pick a number from 1 to n. You have to guess which number I picked. Every time you guess wrong, I will tell you whether the number I picked is higher or lower than your guess. You call a pre-defined API int guess(int num), which returns three possible results:

-1: Your guess is higher than the number I picked (i.e. num > pick). 1: Your guess is lower than the number I picked (i.e. num < pick). 0: your guess is equal to the number I picked (i.e. num == pick).

Return the number that I picked. Example 1: Input: n = 10, pick = 6 Output: 6

Example 2: Input: n = 1, pick = 1 Output: 1

Example 3: Input: n = 2, pick = 1 Output: 1

Constraints:

1 <= n <= 231 - 1 1 <= pick <= n

===
Problem Number: 71 URL: https://leetcode.com/problems/ransom-note Title: 383. Ransom Note Problem Description: Given two strings ransomNote and magazine, return true if ransomNote can be constructed by using the letters from magazine and false otherwise. Each letter in magazine can only be used once in ransomNote. Example 1: Input: ransomNote = "a", magazine = "b" Output: false Example 2: Input: ransomNote = "aa", magazine = "ab" Output: false Example 3: Input: ransomNote = "aa", magazine = "aab" Output: true

Constraints:

1 <= ransomNote.length, magazine.length <= 105 ransomNote and magazine consist of lowercase English letters.

===
Problem Number: 72 URL: https://leetcode.com/problems/first-unique-character-in-a-string Title: 387. First Unique Character in a String Problem Description: Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1. Example 1: Input: s = "leetcode" Output: 0 Example 2: Input: s = "loveleetcode" Output: 2 Example 3: Input: s = "aabb" Output: -1

Constraints:

1 <= s.length <= 105 s consists of only lowercase English letters.

===
Problem Number: 73 URL: https://leetcode.com/problems/find-the-difference Title: 389. Find the Difference Problem Description: You are given two strings s and t. String t is generated by random shuffling string s and then add one more letter at a random position. Return the letter that was added to t.

Example 1: Input: s = ”abcd”, t = ”abcde” Output: ”e” Explanation: ’e’ is the letter that was added.

Example 2: Input: s = ””, t = ”y” Output: ”y”

Constraints:

0 <= s.length <= 1000 t.length == s.length + 1 s and t consist of lowercase English letters.

==

Problem Number: 74 URL: https://leetcode.com/problems/is-subsequence Title: 392. Is Subsequence Problem Description: Given two strings s and t, return true if s is a subsequence of t, or false otherwise. A subsequence of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., ”ace” is a subsequence of ”abcde” while ”aec” is not). Example 1: Input: s = ”abc”, t = ”ahbgdc” Output: true Example 2: Input: s = ”axc”, t = ”ahbgdc” Output: false

Constraints:

0 <= s.length <= 100 0 <= t.length <= 104 s and t consist only of lowercase English letters.

Follow up: Suppose there are lots of incoming s, say s1, s2, ..., sk where k >= 109, and you want to check one by one to see if t has its subsequence. In this scenario, how would you change your code?

==

Problem Number: 75 URL: https://leetcode.com/problems/binary-watch Title: 401. Binary Watch Problem Description: A binary watch has 4 LEDs on the top to represent the hours (0-11), and 6 LEDs on the bottom to represent the minutes (0-59). Each LED represents a zero or one, with the least significant bit on the right.

For example, the below binary watch reads ”4:51”.

Given an integer turnedOn which represents the number of LEDs that are currently on (ignoring the PM), return all possible times the watch could represent. You may return the answer in any order. The hour must not contain a leading zero.

For example, ”01:00” is not valid. It should be ”1:00”.

The minute must consist of two digits and may contain a leading zero.

For example, ”10:2” is not valid. It should be ”10:02”.

Example 1: Input: turnedOn = 1 Output: [”0:01”,”0:02”,”0:04”,”0:08”,”0:16”,”0:32”,”1:00”,”2:00”,”4:00”,”8:0
Example 2: Input: turnedOn = 9 Output: []

Constraints:

0 <= turnedOn <= 10

==
Problem Number: 76 URL: https://leetcode.com/problems/sum-of-left-leaves Title: 404. Sum of Left Leaves Problem Description: Given the root of a binary tree, return the sum of all left leaves. A leaf is a node with no children. A left leaf is a leaf that is the left child of another node. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: 24 Explanation: There are two left leaves in the binary tree, with values 9 and 15 respectively.

Example 2: Input: root = [1] Output: 0

Constraints:

The number of nodes in the tree is in the range [1, 1000]. -1000 <= Node.val <= 1000

==
Problem Number: 77 URL: https://leetcode.com/problems/convert-a-number-to-hexadecimal Title: 405. Convert a Number to Hexadecimal Problem Description: Given an integer num, return a string representing its hexadecimal representation. For negative integers, two's complement method is used. All the letters in the answer string should be lowercase characters, and there should not be any leading zeros in the answer except for the zero itself. Note: You are not allowed to use any built-in library method to directly solve this problem. Example 1: Input: num = 26 Output: "1a" Example 2: Input: num = -1 Output: "ffffffff"

Constraints:

-231 <= num <= 231 - 1

==
Problem Number: 78 URL: https://leetcode.com/problems/longest-palindrome Title: 409. Longest Palindrome Problem Description: Given a string s which consists of lowercase or uppercase letters, return the length of the longest palindrome that can be built with those letters. Letters are case sensitive, for example, "Aa" is not considered a palindrome here. Example 1: Input: s = "abccccdd" Output: 7 Explanation: One longest palindrome that can be built is "dccaccd", whose length is 7.

Example 2: Input: s = "a" Output: 1 Explanation: The longest palindrome that can be built is "a", whose length is 1.

Constraints:

1 <= s.length <= 2000 s consists of lowercase and/or uppercase English letters only.

==
Problem Number: 79 URL: https://leetcode.com/problems/fizz-buzz Title:

412. Fizz Buzz Problem Description: Given an integer n, return a string array answer (1-indexed) where:

answer[i] == "FizzBuzz" if i is divisible by 3 and 5. answer[i] == "Fizz" if i is divisible by 3. answer[i] == "Buzz" if i is divisible by 5. answer[i] == i (as a string) if none of the above conditions are true.

Example 1: Input: n = 3 Output: ["1","2","Fizz"] Example 2: Input: n = 5 Output: ["1","2","Fizz","4","Buzz"] Example 3: Input: n = 15 Output: ["1","2","Fizz","4","Buzz","Fizz","7","8","Fizz","Buzz","11","Fizz","13","14","FizzBuzz"]

Constraints:

1 <= n <= 104

==

Problem Number: 80 URL: https://leetcode.com/problems/third-maximum-number Title: 414. Third Maximum Number Problem Description: Given an integer array nums, return the third distinct maximum number in this array. If the third maximum does not exist, return the maximum number. Example 1: Input: nums = [3,2,1] Output: 1 Explanation: The first distinct maximum is 3. The second distinct maximum is 2. The third distinct maximum is 1.

Example 2: Input: nums = [1,2] Output: 2 Explanation: The first distinct maximum is 2. The second distinct maximum is 1. The third distinct maximum does not exist, so the maximum (2) is returned instead.

Example 3: Input: nums = [2,2,3,1] Output: 1 Explanation: The first distinct maximum is 3. The second distinct maximum is 2 (both 2's are counted together since they have the same value). The third distinct maximum is 1.

Constraints:

1 <= nums.length <= 104 -231 <= nums[i] <= 231 - 1

Follow up: Can you find an O(n) solution? ======================================
Problem Number: 81 URL: https://leetcode.com/problems/add-strings Title: 415. Add Strings Problem Description: Given two non-negative integers, num1 and num2 represented as string, return the sum of num1 and num2 as a string. You must solve the problem without using any built-in library for handling large integers (such as BigInteger). You must also not convert the inputs to integers directly. Example 1: Input: num1 = "11", num2 = "123" Output: "134"

Example 2: Input: num1 = "456", num2 = "77" Output: "533"

Example 3: Input: num1 = "0", num2 = "0" Output: "0"

Constraints:

1 <= num1.length, num2.length <= 104 num1 and num2 consist of only digits. num1 and num2 don't have any leading zeros except for the zero itself.

==
Problem Number: 82 URL: https://leetcode.com/problems/number-of-segments-in-a-string Title: 434. Number of Segments in a String Problem Description: Given a string s, return the number of segments in the string. A segment is defined to be a contiguous sequence of non-space characters. Example 1: Input: s = "Hello, my name is John" Output: 5 Explanation: The five segments are ["Hello,", "my", "name", "is", "John"]

Example 2: Input: s = "Hello" Output: 1

Constraints:

0 <= s.length <= 300 s consists of lowercase and uppercase English letters, digits, or one of the following characters "!@#$%^&*()_+-=',.:". The only space character in s is ' '.

==
Problem Number: 83 URL: https://leetcode.com/problems/arranging-coins Title: 441. Arranging Coins Problem Description: You have n coins and you want to build a staircase with these coins. The staircase consists of k rows where the ith row has exactly i coins. The last row of the staircase may be incomplete. Given the integer n, return the number of complete rows of the staircase you will build. Example 1:

Input: n = 5 Output: 2 Explanation: Because the 3rd row is incomplete, we return 2.

Example 2:

Input: n = 8 Output: 3 Explanation: Because the 4th row is incomplete, we return 3.

Constraints:

1 <= n <= 231 - 1

==
Problem Number: 84 URL: https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array Title: 448. Find All Numbers Disappeared in an Array Problem Description: Given an array nums of n integers where nums[i] is in the range [1, n], return an array of all the integers in the range [1, n] that do not appear in nums. Example 1: Input: nums = [4,3,2,7,8,2,3,1] Output: [5,6] Example 2: Input: nums = [1,1] Output: [2]

Constraints:

n == nums.length 1 <= n <= 105 1 <= nums[i] <= n

Follow up: Could you do it without extra space and in O(n) runtime? You may assume the returned list does not count as extra space.

==
Problem Number: 85 URL: https://leetcode.com/problems/assign-cookies Ti-

tle: 455. Assign Cookies Problem Description: Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie. Each child i has a greed factor g[i], which is the minimum size of a cookie that the child will be content with; and each cookie j has a size s[j]. If s[j] >= g[i], we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.   Example 1: Input: g = [1,2,3], s = [1,1] Output: 1 Explanation: You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3. And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content. You need to output 1.

Example 2: Input: g = [1,2], s = [1,2,3] Output: 2 Explanation: You have 2 children and 3 cookies. The greed factors of 2 children are 1, 2. You have 3 cookies and their sizes are big enough to gratify all of the children, You need to output 2.

Constraints:

1 <= g.length <= 3 * 104 0 <= s.length <= 3 * 104 1 <= g[i], s[j] <= 231 - 1

==
Problem Number: 86 URL: https://leetcode.com/problems/repeated-substring-pattern Title: 459. Repeated Substring Pattern Problem Description: Given a string s, check if it can be constructed by taking a substring of it and appending multiple copies of the substring together.   Example 1: Input: s = "abab" Output: true Explanation: It is the substring "ab" twice.

Example 2: Input: s = "aba" Output: false

Example 3: Input: s = "abcabcabcabc" Output: true Explanation: It is the substring "abc" four times or the substring "abcabc" twice.

Constraints:

1 <= s.length <= 104 s consists of lowercase English letters.

==
Problem Number: 87 URL: https://leetcode.com/problems/hamming-distance Title: 461. Hamming Distance Problem Description: The Hamming distance between two integers is the number of positions at which the corresponding bits are different. Given two integers x and y, return the Hamming distance between them.   Example 1: Input: x = 1, y = 4 Output: 2 Explanation: 1 (0 0 0 1) 4 (0 1 0 0) ↑ ↑ The above arrows point to positions where the corresponding bits are different.

Example 2: Input: x = 3, y = 1 Output: 1

Constraints:

0 <= x, y <= 231 - 1

===
Problem Number: 88 URL: https://leetcode.com/problems/island-perimeter Title: 463. Island Perimeter Problem Description: You are given row x col grid representing a map where grid[i][j] = 1 represents land and grid[i][j] = 0 represents water. Grid cells are connected horizontally/vertically (not diagonally). The grid is completely surrounded by water, and there is exactly one island (i.e., one or more connected land cells). The island doesn't have "lakes", meaning the water inside isn't connected to the water around the island. One cell is a square with side length 1. The grid is rectangular, width and height don't exceed 100. Determine the perimeter of the island. Example 1:

Input: grid = [[0,1,0,0],[1,1,1,0],[0,1,0,0],[1,1,0,0]] Output: 16 Explanation: The perimeter is the 16 yellow stripes in the image above.

Example 2: Input: grid = [[1]] Output: 4

Example 3: Input: grid = [[1,0]] Output: 4

Constraints:

row == grid.length col == grid[i].length 1 <= row, col <= 100 grid[i][j] is 0 or 1. There is exactly one island in grid.

===
Problem Number: 89 URL: https://leetcode.com/problems/number-complement Title: 476. Number Complement Problem Description: The complement of an integer is the integer you get when you flip all the 0's to 1's and all the 1's to 0's in its binary representation.

For example, The integer 5 is "101" in binary and its complement is "010" which is the integer 2.

Given an integer num, return its complement. Example 1: Input: num = 5 Output: 2 Explanation: The binary representation of 5 is 101 (no leading zero bits), and its complement is 010. So you need to output 2.

Example 2: Input: num = 1 Output: 0 Explanation: The binary representation of 1 is 1 (no leading zero bits), and its complement is 0. So you need to output 0.

Constraints:

1 <= num < 231

Note: This question is the same as 1009: https://leetcode.com/problems/complement-of-base-10-integer/

===
Problem Number: 90 URL: https://leetcode.com/problems/license-key-formatting Title: 482. License Key Formatting Problem Description: You are given a license key represented as a string s that consists of only alphanumeric

characters and dashes. The string is separated into n + 1 groups by n dashes. You are also given an integer k. We want to reformat the string s such that each group contains exactly k characters, except for the first group, which could be shorter than k but still must contain at least one character. Furthermore, there must be a dash inserted between two groups, and you should convert all lowercase letters to uppercase. Return the reformatted license key. Example 1: Input: s = "5F3Z-2e-9-w", k = 4 Output: "5F3Z-2E9W" Explanation: The string s has been split into two parts, each part has 4 characters. Note that the two extra dashes are not needed and can be removed.

Example 2: Input: s = "2-5g-3-J", k = 2 Output: "2-5G-3J" Explanation: The string s has been split into three parts, each part has 2 characters except the first part as it could be shorter as mentioned above.

Constraints:

1 <= s.length <= 105 s consists of English letters, digits, and dashes '-'. 1 <= k <= 104

==
Problem Number: 91 URL: https://leetcode.com/problems/max-consecutive-ones Title: 485. Max Consecutive Ones Problem Description: Given a binary array nums, return the maximum number of consecutive 1's in the array. Example 1: Input: nums = [1,1,0,1,1,1] Output: 3 Explanation: The first two digits or the last three digits are consecutive 1s. The maximum number of consecutive 1s is 3.

Example 2: Input: nums = [1,0,1,1,0,1] Output: 2

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1.

==
Problem Number: 92 URL: https://leetcode.com/problems/construct-the-rectangle Title: 492. Construct the Rectangle Problem Description: A web developer needs to know how to design a web page's size. So, given a specific rectangular web page's area, your job by now is to design a rectangular web page, whose length L and width W satisfy the following requirements:

The area of the rectangular web page you designed must equal to the given target area. The width W should not be larger than the length L, which means L >= W. The difference between length L and width W should be as small as possible.

Return an array [L, W] where L and W are the length and width of the web page you designed in sequence. Example 1: Input: area = 4 Output: [2,2] Explanation: The target area is 4, and all the possible ways to construct it are [1,4], [2,2], [4,1]. But according to requirement 2, [1,4] is illegal; according to requirement 3, [4,1] is not optimal compared to [2,2]. So the length L is 2, and the width W is 2.

Example 2: Input: area = 37 Output: [37,1]

Example 3: Input: area = 122122 Output: [427,286]

Constraints:

1 <= area <= 107

===

Problem Number: 93 URL: https://leetcode.com/problems/teemo-attacking Title: 495. Teemo Attacking Problem Description: Our hero Teemo is attacking an enemy Ashe with poison attacks! When Teemo attacks Ashe, Ashe gets poisoned for a exactly duration seconds. More formally, an attack at second t will mean Ashe is poisoned during the inclusive time interval [t, t + duration - 1]. If Teemo attacks again before the poison effect ends, the timer for it is reset, and the poison effect will end duration seconds after the new attack. You are given a non-decreasing integer array timeSeries, where timeSeries[i] denotes that Teemo attacks Ashe at second timeSeries[i], and an integer duration. Return the total number of seconds that Ashe is poisoned. Example 1: Input: timeSeries = [1,4], duration = 2 Output: 4 Explanation: Teemo's attacks on Ashe go as follows: - At second 1, Teemo attacks, and Ashe is poisoned for seconds 1 and 2. - At second 4, Teemo attacks, and Ashe is poisoned for seconds 4 and 5. Ashe is poisoned for seconds 1, 2, 4, and 5, which is 4 seconds in total.

Example 2: Input: timeSeries = [1,2], duration = 2 Output: 3 Explanation: Teemo's attacks on Ashe go as follows: - At second 1, Teemo attacks, and Ashe is poisoned for seconds 1 and 2. - At second 2 however, Teemo attacks again and resets the poison timer. Ashe is poisoned for seconds 2 and 3. Ashe is poisoned for seconds 1, 2, and 3, which is 3 seconds in total. Constraints:

1 <= timeSeries.length <= 104 0 <= timeSeries[i], duration <= 107 timeSeries is sorted in non-decreasing order.

===

Problem Number: 94 URL: https://leetcode.com/problems/next-greater-element-i Title: 496. Next Greater Element I Problem Description: The next greater element of some element x in an array is the first greater element that is to the right of x in the same array. You are given two distinct 0-indexed integer arrays nums1 and nums2, where nums1 is a subset of nums2. For each 0 <= i < nums1.length, find the index j such that nums1[i] == nums2[j] and determine the next greater element of nums2[j] in nums2. If there is no next greater element, then the answer for this query is -1. Return an array ans of length nums1.length such that ans[i] is the next greater element as described above. Example 1: Input: nums1 = [4,1,2], nums2 = [1,3,4,2] Output: [-1,3,-1] Explanation: The next greater element for each value of nums1 is as follows: - 4 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1. - 1 is underlined in nums2 = [1,3,4,2]. The next greater element is 3. - 2 is underlined in nums2 = [1,3,4,2]. There is no next greater element, so the answer is -1.

Example 2: Input: nums1 = [2,4], nums2 = [1,2,3,4] Output: [3,-1] Explanation: The next greater element for each value of nums1 is as follows: - 2 is underlined in nums2 = [1,2,3,4]. The next greater element is 3. - 4 is underlined in nums2 = [1,2,3,4]. There is no next greater element, so the answer is -1.

Constraints:

1 <= nums1.length <= nums2.length <= 1000 0 <= nums1[i], nums2[i] <= 104 All integers in nums1 and nums2 are unique. All the integers of nums1 also appear in nums2.

Follow up: Could you find an O(nums1.length + nums2.length) solution?

==

Problem Number: 95 URL: https://leetcode.com/problems/keyboard-row Title: 500. Keyboard Row Problem Description: Given an array of strings words, return the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below. In the American keyboard:

the first row consists of the characters "qwertyuiop", the second row consists of the characters "asdfghjkl", and the third row consists of the characters "zxcvbnm".

Example 1: Input: words = ["Hello","Alaska","Dad","Peace"] Output: ["Alaska","Dad"]

Example 2: Input: words = ["omk"] Output: []

Example 3: Input: words = ["adsdf","sfd"] Output: ["adsdf","sfd"]

Constraints:

1 <= words.length <= 20 1 <= words[i].length <= 100 words[i] consists of English letters (both lowercase and uppercase).

==

Problem Number: 96 URL: https://leetcode.com/problems/find-mode-in-binary-search-tree Title: 501. Find Mode in Binary Search Tree Problem Description: Given the root of a binary search tree (BST) with duplicates, return all the mode(s) (i.e., the most frequently occurred element) in it. If the tree has more than one mode, return them in any order. Assume a BST is defined as follows:

The left subtree of a node contains only nodes with keys less than or equal to the node's key. The right subtree of a node contains only nodes with keys greater than or equal to the node's key. Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [1,null,2,2] Output: [2]

Example 2: Input: root = [0] Output: [0]

Constraints:

The number of nodes in the tree is in the range [1, 104]. -105 <= Node.val <= 105

Follow up: Could you do that without using any extra space? (Assume that the implicit stack space incurred due to recursion does not count). ==

Problem Number: 97 URL: https://leetcode.com/problems/base-7 Title: 504. Base 7 Problem Description: Given an integer num, return a string of its base 7 representation. Example 1: Input: num = 100 Output: "202" Example 2: Input: num = -7 Output: "-10"

Constraints:

-107 <= num <= 107

==

Problem Number: 98 URL: https://leetcode.com/problems/relative-ranks Title: 506. Relative Ranks Problem Description: You are given an integer array score of size n, where score[i] is the score of the ith athlete in a competition. All the scores are guaranteed to be unique. The athletes are placed based on their scores, where the 1st place athlete has the highest score, the 2nd place athlete has the 2nd highest score, and so on. The placement of each athlete determines their rank:

The 1st place athlete's rank is "Gold Medal". The 2nd place athlete's rank is "Silver Medal". The 3rd place athlete's rank is "Bronze Medal". For the 4th place to the nth place athlete, their rank is their placement number (i.e., the xth place athlete's rank is "x").

Return an array answer of size n where answer[i] is the rank of the ith athlete. Example 1: Input: score = [5,4,3,2,1] Output: ["Gold Medal","Silver Medal","Bronze Medal","4","5"] Explanation: The placements are [1st, 2nd, 3rd, 4th, 5th]. Example 2: Input: score = [10,3,8,9,4] Output: ["Gold Medal","5","Bronze Medal","Silver Medal","4"] Explanation: The placements are [1st, 5th, 3rd, 2nd, 4th].

Constraints:

n == score.length 1 <= n <= 104 0 <= score[i] <= 106 All the values in score are unique.

==

Problem Number: 99 URL: https://leetcode.com/problems/perfect-number Title: 507. Perfect Number Problem Description: A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. A divisor of an integer x is an integer that can divide x evenly. Given an integer n, return true if n is a perfect number, otherwise return false.

Example 1: Input: num = 28 Output: true Explanation: 28 = 1 + 2 + 4 + 7 + 14 1, 2, 4, 7, and 14 are all divisors of 28.

Example 2: Input: num = 7 Output: false

Constraints:

1 <= num <= 108

==

Problem Number: 100 URL: https://leetcode.com/problems/fibonacci-number Title: 509. Fibonacci Number Problem Description: The Fibonacci numbers, commonly denoted F(n) form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is, F(0) = 0, F(1) = 1 F(n) = F(n - 1) + F(n - 2), for n > 1.

Given n, calculate F(n). Example 1: Input: n = 2 Output: 1 Explanation: F(2) = F(1) + F(0) = 1 + 0 = 1.

Example 2: Input: n = 3 Output: 2 Explanation: F(3) = F(2) + F(1) = 1 + 1 = 2.

Example 3: Input: n = 4 Output: 3 Explanation: F(4) = F(3) + F(2) = 2 + 1 = 3.

Constraints:

0 <= n <= 30

==

Problem Number: 101 URL: https://leetcode.com/problems/detect-capital Title: 520. Detect Capital Problem Description: We define the usage of capitals in a word to be right when one of the following cases holds:

All letters in this word are capitals, like "USA". All letters in this word are not capitals, like "leetcode". Only the first letter in this word is capital, like "Google".

Given a string word, return true if the usage of capitals in it is right. Example 1: Input: word = "USA" Output: true Example 2: Input: word = "FlaG" Output: false

Constraints:

1 <= word.length <= 100 word consists of lowercase and uppercase English letters.

==

Problem Number: 102 URL: https://leetcode.com/problems/longest-uncommon-subsequence-i Title: 521. Longest Uncommon Subsequence I Problem Description: Given two strings a and b, return the length of the longest uncommon subsequence between a and b. If the longest uncommon subsequence does not exist, return -1. An uncommon subsequence between two strings is a string that is a subsequence of one but not the other. A subsequence of a string s is a string that can be obtained after deleting any number of characters from s.

For example, "abc" is a subsequence of "aebdc" because you can delete the underlined characters in "aebdc" to get "abc". Other subsequences of "aebdc" include "aebdc", "aeb", and "" (empty string).

Example 1: Input: a = "aba", b = "cdc" Output: 3 Explanation: One longest uncommon subsequence is "aba" because "aba" is a subsequence of "aba" but not "cdc". Note that "cdc" is also a longest uncommon subsequence.

Example 2: Input: a = "aaa", b = "bbb" Output: 3 Explanation: The longest uncommon subsequences are "aaa" and "bbb".

Example 3: Input: a = "aaa", b = "aaa" Output: -1 Explanation: Every subsequence of string a is also a subsequence of string b. Similarly, every subsequence of string b is also a subsequence of string a.

Constraints:

1 <= a.length, b.length <= 100 a and b consist of lower-case English letters.

==
Problem Number: 103 URL: https://leetcode.com/problems/minimum-absolute-difference-in-bst Title: 530. Minimum Absolute Difference in BST Problem Description: Given the root of a Binary Search Tree (BST), return the minimum absolute difference between the values of any two different nodes in the tree. Example 1:

Input: root = [4,2,6,1,3] Output: 1

Example 2:

Input: root = [1,0,48,null,null,12,49] Output: 1

Constraints:

The number of nodes in the tree is in the range [2, 104]. 0 <= Node.val <= 105

Note: This question is the same as 783: https://leetcode.com/problems/minimum-distance-between-bst-nodes/

==
Problem Number: 104 URL: https://leetcode.com/problems/reverse-string-ii Title: 541. Reverse String II Problem Description: Given a string s and an integer k, reverse the first k characters for every 2k characters counting from the start of the string. If there are fewer than k characters left, reverse all of them. If there are less than 2k but greater than or equal to k characters, then reverse the first k characters and leave the other as original. Example 1: Input: s = "abcdefg", k = 2 Output: "bacdfeg" Example 2: Input: s = "abcd", k = 2 Output: "bacd"

Constraints:

1 <= s.length <= 104 s consists of only lowercase English letters. 1 <= k <= 104

==
Problem Number: 105 URL: https://leetcode.com/problems/diameter-of-binary-tree Title: 543. Diameter of Binary Tree Problem Description: Given the root of a binary tree, return the length of the diameter of the tree. The diameter of a binary tree is the length of the longest path between any two nodes in a tree. This path may or may not pass through the root. The length of a path between two nodes is represented by the number of edges between them. Example 1:

Input: root = [1,2,3,4,5] Output: 3 Explanation: 3 is the length of the path [4,2,1,3] or [5,2,1,3].

Example 2: Input: root = [1,2] Output: 1

Constraints:

The number of nodes in the tree is in the range [1, 104]. -100 <= Node.val <= 100

==
Problem Number: 106 URL: https://leetcode.com/problems/student-attendance-record-i Title: 551. Student Attendance Record I Problem Description: You are given a string s representing an attendance record for a student where each character signifies whether the student was absent, late, or present on that day. The record only contains the following three characters:

'A': Absent. 'L': Late. 'P': Present.

The student is eligible for an attendance award if they meet both of the following criteria:

The student was absent ('A') for strictly fewer than 2 days total. The student was never late ('L') for 3 or more consecutive days.

Return true if the student is eligible for an attendance award, or false otherwise. Example 1: Input: s = "PPALLP" Output: true Explanation: The student has fewer than 2 absences and was never late 3 or more consecutive days.

Example 2: Input: s = "PPALLL" Output: false Explanation: The student was late 3 consecutive days in the last 3 days, so is not eligible for the award.

Constraints:

1 <= s.length <= 1000 s[i] is either 'A', 'L', or 'P'.

==
Problem Number: 107 URL: https://leetcode.com/problems/reverse-words-in-a-string-iii Title: 557. Reverse Words in a String III Problem Description:

Given a string s, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order. Example 1: Input: s = "Let's take LeetCode contest" Output: "s'teL ekat edoCteeL tsetnoc" Example 2: Input: s = "God Ding" Output: "doG gniD"

Constraints:

1 <= s.length <= 5 * 104 s contains printable ASCII characters. s does not contain any leading or trailing spaces. There is at least one word in s. All the words in s are separated by a single space.

==
Problem Number: 108 URL: https://leetcode.com/problems/maximum-depth-of-n-ary-tree Title: 559. Maximum Depth of N-ary Tree Problem Description: Given a n-ary tree, find its maximum depth. The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node. Nary-Tree input serialization is represented in their level order traversal, each group of children is separated by the null value (See examples). Example 1:

Input: root = [1,null,3,2,4,null,5,6] Output: 3

Example 2:

Input: root = [1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14] Output: 5

Constraints:

The total number of nodes is in the range [0, 104]. The depth of the n-ary tree is less than or equal to 1000.

==
Problem Number: 109 URL: https://leetcode.com/problems/array-partition Title: 561. Array Partition Problem Description: Given an integer array nums of 2n integers, group these integers into n pairs (a1, b1), (a2, b2), ..., (an, bn) such that the sum of min(ai, bi) for all i is maximized. Return the maximized sum. Example 1: Input: nums = [1,4,3,2] Output: 4 Explanation: All possible pairings (ignoring the ordering of elements) are: 1. (1, 4), (2, 3) -> min(1, 4) + min(2, 3) = 1 + 2 = 3 2. (1, 3), (2, 4) -> min(1, 3) + min(2, 4) = 1 + 2 = 3 3. (1, 2), (3, 4) -> min(1, 2) + min(3, 4) = 1 + 3 = 4 So the maximum possible sum is 4. Example 2: Input: nums = [6,2,6,5,1,2] Output: 9 Explanation: The optimal pairing is (2, 1), (2, 5), (6, 6). min(2, 1) + min(2, 5) + min(6, 6) = 1 + 2 + 6 = 9.

Constraints:

1 <= n <= 104 nums.length == 2 * n -104 <= nums[i] <= 104

==
Problem Number: 110 URL: https://leetcode.com/problems/binary-tree-tilt Title: 563. Binary Tree Tilt Problem Description: Given the root of a binary

tree, return the sum of every tree node's tilt. The tilt of a tree node is the absolute difference between the sum of all left subtree node values and all right subtree node values. If a node does not have a left child, then the sum of the left subtree node values is treated as 0. The rule is similar if the node does not have a right child.   Example 1:

Input: root = [1,2,3] Output: 1 Explanation: Tilt of node 2 : |0-0| = 0 (no children) Tilt of node 3 : |0-0| = 0 (no children) Tilt of node 1 : |2-3| = 1 (left subtree is just left child, so sum is 2; right subtree is just right child, so sum is 3) Sum of every tilt : 0 + 0 + 1 = 1

Example 2:

Input: root = [4,2,9,3,5,null,7] Output: 15 Explanation: Tilt of node 3 : |0-0| = 0 (no children) Tilt of node 5 : |0-0| = 0 (no children) Tilt of node 7 : |0-0| = 0 (no children) Tilt of node 2 : |3-5| = 2 (left subtree is just left child, so sum is 3; right subtree is just right child, so sum is 5) Tilt of node 9 : |0-7| = 7 (no left child, so sum is 0; right subtree is just right child, so sum is 7) Tilt of node 4 : |(3+5+2)-(9+7)| = |10-16| = 6 (left subtree values are 3, 5, and 2, which sums to 10; right subtree values are 9 and 7, which sums to 16) Sum of every tilt : 0 + 0 + 0 + 2 + 7 + 6 = 15

Example 3:

Input: root = [21,7,14,1,1,2,2,3,3] Output: 9

Constraints:

The number of nodes in the tree is in the range [0, 104]. -1000 <= Node.val <= 1000

==
Problem Number: 111 URL: https://leetcode.com/problems/reshape-the-matrix Title: 566. Reshape the Matrix Problem Description: In MATLAB, there is a handy function called reshape which can reshape an m x n matrix into a new one with a different size r x c keeping its original data. You are given an m x n matrix mat and two integers r and c representing the number of rows and the number of columns of the wanted reshaped matrix. The reshaped matrix should be filled with all the elements of the original matrix in the same row-traversing order as they were. If the reshape operation with given parameters is possible and legal, output the new reshaped matrix; Otherwise, output the original matrix.   Example 1:

Input: mat = [[1,2],[3,4]], r = 1, c = 4 Output: [[1,2,3,4]]

Example 2:

Input: mat = [[1,2],[3,4]], r = 2, c = 4 Output: [[1,2],[3,4]]

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 100 -1000 <= mat[i][j] <= 1000 1 <= r, c <= 300

==
Problem Number: 112 URL: https://leetcode.com/problems/subtree-of-another-tree Title: 572. Subtree of Another Tree Problem Description: Given the roots of two binary trees root and subRoot, return true if there is a subtree of root with the same structure and node values of subRoot and false otherwise. A subtree of a binary tree tree is a tree that consists of a node in tree and all of this node's descendants. The tree tree could also be considered as a subtree of itself. Example 1:

Input: root = [3,4,5,1,2], subRoot = [4,1,2] Output: true

Example 2:

Input: root = [3,4,5,1,2,null,null,null,null,0], subRoot = [4,1,2] Output: false

Constraints:

The number of nodes in the root tree is in the range [1, 2000]. The number of nodes in the subRoot tree is in the range [1, 1000]. -104 <= root.val <= 104 -104 <= subRoot.val <= 104

==
Problem Number: 113 URL: https://leetcode.com/problems/distribute-candies Title: 575. Distribute Candies Problem Description: Alice has n candies, where the ith candy is of type candyType[i]. Alice noticed that she started to gain weight, so she visited a doctor. The doctor advised Alice to only eat n / 2 of the candies she has (n is always even). Alice likes her candies very much, and she wants to eat the maximum number of different types of candies while still following the doctor's advice. Given the integer array candyType of length n, return the maximum number of different types of candies she can eat if she only eats n / 2 of them. Example 1: Input: candyType = [1,1,2,2,3,3] Output: 3 Explanation: Alice can only eat 6 / 2 = 3 candies. Since there are only 3 types, she can eat one of each type.

Example 2: Input: candyType = [1,1,2,3] Output: 2 Explanation: Alice can only eat 4 / 2 = 2 candies. Whether she eats types [1,2], [1,3], or [2,3], she still can only eat 2 different types.

Example 3: Input: candyType = [6,6,6,6] Output: 1 Explanation: Alice can only eat 4 / 2 = 2 candies. Even though she can eat 2 candies, she only has 1 type.

Constraints:

n == candyType.length 2 <= n <= 104 n is even. -105 <= candyType[i] <= 105

==
Problem Number: 114 URL: https://leetcode.com/problems/n-ary-tree-

preorder-traversal Title: 589. N-ary Tree Preorder Traversal Problem Description: Given the root of an n-ary tree, return the preorder traversal of its nodes' values. Nary-Tree input serialization is represented in their level order traversal. Each group of children is separated by the null value (See examples) Example 1:

Input: root = [1,null,3,2,4,null,5,6] Output: [1,3,5,6,2,4]

Example 2:

Input: root = [1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14] Output: [1,2,3,6,7,11,14,4,8,12,5,9,13,10]

Constraints:

The number of nodes in the tree is in the range [0, 104]. 0 <= Node.val <= 104 The height of the n-ary tree is less than or equal to 1000.

Follow up: Recursive solution is trivial, could you do it iteratively?

==
Problem Number: 115 URL: https://leetcode.com/problems/n-ary-tree-postorder-traversal Title: 590. N-ary Tree Postorder Traversal Problem Description: Given the root of an n-ary tree, return the postorder traversal of its nodes' values. Nary-Tree input serialization is represented in their level order traversal. Each group of children is separated by the null value (See examples) Example 1:

Input: root = [1,null,3,2,4,null,5,6] Output: [5,6,3,2,4,1]

Example 2:

Input: root = [1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14] Output: [2,6,14,11,7,3,12,8,4,13,9,10,5,1]

Constraints:

The number of nodes in the tree is in the range [0, 104]. 0 <= Node.val <= 104 The height of the n-ary tree is less than or equal to 1000.

Follow up: Recursive solution is trivial, could you do it iteratively?

==
Problem Number: 116 URL: https://leetcode.com/problems/longest-harmonious-subsequence Title: 594. Longest Harmonious Subsequence Problem Description: We define a harmonious array as an array where the difference between its maximum value and its minimum value is exactly 1. Given an integer array nums, return the length of its longest harmonious subsequence among all its possible subsequences. A subsequence of array is a sequence that can be derived from the array by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: nums = [1,3,2,2,5,2,3,7] Output: 5 Explanation: The longest harmonious subsequence is [3,2,2,2,3].

Example 2: Input: nums = [1,2,3,4] Output: 2

Example 3: Input: nums = [1,1,1,1] Output: 0

Constraints:

1 <= nums.length <= 2 * 104 -109 <= nums[i] <= 109

==

Problem Number: 117 URL: https://leetcode.com/problems/range-addition-ii Title: 598. Range Addition II Problem Description: You are given an m x n matrix M initialized with all 0's and an array of operations ops, where ops[i] = [ai, bi] means M[x][y] should be incremented by one for all 0 <= x < ai and 0 <= y < bi. Count and return the number of maximum integers in the matrix after performing all the operations. Example 1:

Input: m = 3, n = 3, ops = [[2,2],[3,3]] Output: 4 Explanation: The maximum integer in M is 2, and there are four of it in M. So return 4.

Example 2: Input: m = 3, n = 3, ops = [[2,2],[3,3],[3,3],[3,3],[2,2],[3,3],[3,3],[3,3],[2,2],[3,3],[3,3],[3,3]] Output: 4

Example 3: Input: m = 3, n = 3, ops = [] Output: 9

Constraints:

1 <= m, n <= 4 * 104 0 <= ops.length <= 104 ops[i].length == 2 1 <= ai <= m 1 <= bi <= n

==

Problem Number: 118 URL: https://leetcode.com/problems/minimum-index-sum-of-two-lists Title: 599. Minimum Index Sum of Two Lists Problem Description: Given two arrays of strings list1 and list2, find the common strings with the least index sum. A common string is a string that appeared in both list1 and list2. A common string with the least index sum is a common string such that if it appeared at list1[i] and list2[j] then i + j should be the minimum value among all the other common strings. Return all the common strings with the least index sum. Return the answer in any order. Example 1: Input: list1 = ["Shogun","Tapioca Express","Burger King","KFC"], list2 = ["Piatti","The Grill at Torrey Pines","Hungry Hunter Steakhouse","Shogun"] Output: ["Shogun"] Explanation: The only common string is "Shogun".

Example 2: Input: list1 = ["Shogun","Tapioca Express","Burger King","KFC"], list2 = ["KFC","Shogun","Burger King"] Output: ["Shogun"] Explanation: The common string with the least index sum is "Shogun" with index sum = (0 + 1) = 1.

Example 3: Input: list1 = ["happy","sad","good"], list2 = ["sad","happy","good"] Output: ["sad","happy"] Explanation: There are three common strings: "happy" with index sum = (0 + 1) = 1. "sad" with index sum = (1 + 0) = 1. "good" with index sum = (2 + 2) = 4. The strings with the least index sum are "sad" and "happy".

Constraints:

1 <= list1.length, list2.length <= 1000 1 <= list1[i].length, list2[i].length <= 30 list1[i] and list2[i] consist of spaces ' ' and English letters. All the strings of list1 are unique. All the strings of list2 are unique. There is at least a common string between list1 and list2.

==
Problem Number: 119 URL: https://leetcode.com/problems/can-place-flowers Title: 605. Can Place Flowers Problem Description: You have a long flowerbed in which some of the plots are planted, and some are not. However, flowers cannot be planted in adjacent plots. Given an integer array flowerbed containing 0's and 1's, where 0 means empty and 1 means not empty, and an integer n, return true if n new flowers can be planted in the flowerbed without violating the no-adjacent-flowers rule and false otherwise. Example 1: Input: flowerbed = [1,0,0,0,1], n = 1 Output: true Example 2: Input: flowerbed = [1,0,0,0,1], n = 2 Output: false

Constraints:

1 <= flowerbed.length <= 2 * 104 flowerbed[i] is 0 or 1. There are no two adjacent flowers in flowerbed. 0 <= n <= flowerbed.length

==
Problem Number: 120 URL: https://leetcode.com/problems/construct-string-from-binary-tree Title: 606. Construct String from Binary Tree Problem Description: Given the root of a binary tree, construct a string consisting of parenthesis and integers from a binary tree with the preorder traversal way, and return it. Omit all the empty parenthesis pairs that do not affect the one-to-one mapping relationship between the string and the original binary tree. Example 1:

Input: root = [1,2,3,4] Output: "1(2(4))(3)" Explanation: Originally, it needs to be "1(2(4)())(3()())", but you need to omit all the unnecessary empty parenthesis pairs. And it will be "1(2(4))(3)"

Example 2:

Input: root = [1,2,3,null,4] Output: "1(2()(4))(3)" Explanation: Almost the same as the first example, except we cannot omit the first parenthesis pair to break the one-to-one mapping relationship between the input and the output.

Constraints:

The number of nodes in the tree is in the range [1, 104]. -1000 <= Node.val <= 1000

==
Problem Number: 121 URL: https://leetcode.com/problems/merge-two-binary-trees Title: 617. Merge Two Binary Trees Problem Description: You are given two binary trees root1 and root2. Imagine that when you put one of them to

cover the other, some nodes of the two trees are overlapped while the others are not. You need to merge the two trees into a new binary tree. The merge rule is that if two nodes overlap, then sum node values up as the new value of the merged node. Otherwise, the NOT null node will be used as the node of the new tree. Return the merged tree. Note: The merging process must start from the root nodes of both trees. Example 1:

Input: root1 = [1,3,2,5], root2 = [2,1,3,null,4,null,7] Output: [3,4,5,5,4,null,7]

Example 2: Input: root1 = [1], root2 = [1,2] Output: [2,2]

Constraints:

The number of nodes in both trees is in the range [0, 2000]. -104 <= Node.val <= 104

==

Problem Number: 122 URL: https://leetcode.com/problems/maximum-product-of-three-numbers Title: 628. Maximum Product of Three Numbers Problem Description: Given an integer array nums, find three numbers whose product is maximum and return the maximum product. Example 1: Input: nums = [1,2,3] Output: 6 Example 2: Input: nums = [1,2,3,4] Output: 24 Example 3: Input: nums = [-1,-2,-3] Output: -6

Constraints:

3 <= nums.length <= 104 -1000 <= nums[i] <= 1000

==

Problem Number: 123 URL: https://leetcode.com/problems/average-of-levels-in-binary-tree Title: 637. Average of Levels in Binary Tree Problem Description: Given the root of a binary tree, return the average value of the nodes on each level in the form of an array. Answers within 10-5 of the actual answer will be accepted. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: [3.00000,14.50000,11.00000] Explanation: The average value of nodes on level 0 is 3, on level 1 is 14.5, and on level 2 is 11. Hence return [3, 14.5, 11].

Example 2:

Input: root = [3,9,20,15,7] Output: [3.00000,14.50000,11.00000]

Constraints:

The number of nodes in the tree is in the range [1, 104]. -231 <= Node.val <= 231 - 1

==

Problem Number: 124 URL: https://leetcode.com/problems/maximum-average-subarray-i Title: 643. Maximum Average Subarray I Problem Description: You are given an integer array nums consisting of n elements, and an integer k. Find a contiguous subarray whose length is equal to k that has the maximum average value and return this value. Any answer with a calculation error less than 10-5 will be accepted. Example 1: Input: nums = [1,12,-5,-6,50,3], k = 4 Output: 12.75000 Explanation: Maximum average is (12 - 5 - 6 + 50) / 4 = 51 / 4 = 12.75

Example 2: Input: nums = [5], k = 1 Output: 5.00000

Constraints:

n == nums.length 1 <= k <= n <= 105 -104 <= nums[i] <= 104

==
Problem Number: 125 URL: https://leetcode.com/problems/set-mismatch Title: 645. Set Mismatch Problem Description: You have a set of integers s, which originally contains all the numbers from 1 to n. Unfortunately, due to some error, one of the numbers in s got duplicated to another number in the set, which results in repetition of one number and loss of another number. You are given an integer array nums representing the data status of this set after the error. Find the number that occurs twice and the number that is missing and return them in the form of an array. Example 1: Input: nums = [1,2,2,4] Output: [2,3] Example 2: Input: nums = [1,1] Output: [1,2]

Constraints:

2 <= nums.length <= 104 1 <= nums[i] <= 104

==
Problem Number: 126 URL: https://leetcode.com/problems/two-sum-iv-input-is-a-bst Title: 653. Two Sum IV - Input is a BST Problem Description: Given the root of a binary search tree and an integer k, return true if there exist two elements in the BST such that their sum is equal to k, or false otherwise. Example 1:

Input: root = [5,3,6,2,4,null,7], k = 9 Output: true

Example 2:

Input: root = [5,3,6,2,4,null,7], k = 28 Output: false

Constraints:

The number of nodes in the tree is in the range [1, 104]. -104 <= Node.val <= 104 root is guaranteed to be a valid binary search tree. -105 <= k <= 105

==
Problem Number: 127 URL: https://leetcode.com/problems/robot-return-to-origin Title: 657. Robot Return to Origin Problem Description: There is a robot starting at the position (0, 0), the origin, on a 2D plane. Given a sequence of its moves, judge if this robot ends up at (0, 0) after it completes its moves. You are given a string moves that represents the move sequence of the robot where moves[i] represents its ith move. Valid moves are 'R' (right), 'L' (left), 'U' (up), and 'D' (down). Return true if the robot returns to the origin

after it finishes all of its moves, or false otherwise. Note: The way that the robot is ”facing” is irrelevant. 'R' will always make the robot move to the right once, 'L' will always make it move left, etc. Also, assume that the magnitude of the robot's movement is the same for each move. Example 1: Input: moves = ”UD” Output: true Explanation: The robot moves up once, and then down once. All moves have the same magnitude, so it ended up at the origin where it started. Therefore, we return true.

Example 2: Input: moves = ”LL” Output: false Explanation: The robot moves left twice. It ends up two ”moves” to the left of the origin. We return false because it is not at the origin at the end of its moves.

Constraints:

1 <= moves.length <= 2 * 104 moves only contains the characters 'U', 'D', 'L' and 'R'.

===
Problem Number: 128 URL: https://leetcode.com/problems/image-smoother Title: 661. Image Smoother Problem Description: An image smoother is a filter of the size 3 x 3 that can be applied to each cell of an image by rounding down the average of the cell and the eight surrounding cells (i.e., the average of the nine cells in the blue smoother). If one or more of the surrounding cells of a cell is not present, we do not consider it in the average (i.e., the average of the four cells in the red smoother).

Given an m x n integer matrix img representing the grayscale of an image, return the image after applying the smoother on each cell of it. Example 1:

Input: img = [[1,1,1],[1,0,1],[1,1,1]] Output: [[0,0,0],[0,0,0],[0,0,0]] Explanation: For the points (0,0), (0,2), (2,0), (2,2): floor(3/4) = floor(0.75) = 0 For the points (0,1), (1,0), (1,2), (2,1): floor(5/6) = floor(0.83333333) = 0 For the point (1,1): floor(8/9) = floor(0.88888889) = 0

Example 2:

Input: img = [[100,200,100],[200,50,200],[100,200,100]] Output: [[137,141,137],[141,138,141],[137,141,137]] Explanation: For the points (0,0), (0,2), (2,0), (2,2): floor((100+200+200+50)/4) = floor(137.5) = 137 For the points (0,1), (1,0), (1,2), (2,1): floor((200+200+50+200+100+100)/6) = floor(141.666667) = 141 For the point (1,1): floor((50+200+200+200+200+100+100+100+100)/9) = floor(138.888889) = 138

Constraints:

m == img.length n == img[i].length 1 <= m, n <= 200 0 <= img[i][j] <= 255

===
Problem Number: 129 URL: https://leetcode.com/problems/second-minimum-node-in-a-binary-tree Title: 671. Second Minimum Node In a Binary Tree Problem Description: Given a non-empty special binary tree consisting of nodes with the non-negative value, where each node in this tree has exactly two

or zero sub-node. If the node has two sub-nodes, then this node's value is the smaller value among its two sub-nodes. More formally, the property root.val = min(root.left.val, root.right.val) always holds. Given such a binary tree, you need to output the second minimum value in the set made of all the nodes' value in the whole tree. If no such second minimum value exists, output -1 instead. Example 1:

Input: root = [2,2,5,null,null,5,7] Output: 5 Explanation: The smallest value is 2, the second smallest value is 5.

Example 2:

Input: root = [2,2,2] Output: -1 Explanation: The smallest value is 2, but there isn't any second smallest value.

Constraints:

The number of nodes in the tree is in the range [1, 25]. 1 <= Node.val <= 231 - 1 root.val == min(root.left.val, root.right.val) for each internal node of the tree.

==
Problem Number: 130 URL: https://leetcode.com/problems/longest-continuous-increasing-subsequence Title: 674. Longest Continuous Increasing Subsequence Problem Description: Given an unsorted array of integers nums, return the length of the longest continuous increasing subsequence (i.e. subarray). The subsequence must be strictly increasing. A continuous increasing subsequence is defined by two indices l and r (l < r) such that it is [nums[l], nums[l + 1], ..., nums[r - 1], nums[r]] and for each l <= i < r, nums[i] < nums[i + 1]. Example 1: Input: nums = [1,3,5,4,7] Output: 3 Explanation: The longest continuous increasing subsequence is [1,3,5] with length 3. Even though [1,3,5,7] is an increasing subsequence, it is not continuous as elements 5 and 7 are separated by element 4.

Example 2: Input: nums = [2,2,2,2,2] Output: 1 Explanation: The longest continuous increasing subsequence is [2] with length 1. Note that it must be strictly increasing.

Constraints:

1 <= nums.length <= 104 -109 <= nums[i] <= 109

==
Problem Number: 131 URL: https://leetcode.com/problems/valid-palindrome-ii Title: 680. Valid Palindrome II Problem Description: Given a string s, return true if the s can be palindrome after deleting at most one character from it. Example 1: Input: s = "aba" Output: true

Example 2: Input: s = "abca" Output: true Explanation: You could delete the character 'c'.

Example 3: Input: s = "abc" Output: false

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters.

==
Problem Number: 132 URL: https://leetcode.com/problems/baseball-game Title: 682. Baseball Game Problem Description: You are keeping the scores for a baseball game with strange rules. At the beginning of the game, you start with an empty record. You are given a list of strings operations, where operations[i] is the ith operation you must apply to the record and is one of the following:

An integer x.

Record a new score of x.

'+'.

Record a new score that is the sum of the previous two scores.

'D'.

Record a new score that is the double of the previous score.

'C'.

Invalidate the previous score, removing it from the record.

Return the sum of all the scores on the record after applying all the operations. The test cases are generated such that the answer and all intermediate calculations fit in a 32-bit integer and that all operations are valid. Example 1: Input: ops = ["5","2","C","D","+"] Output: 30 Explanation: "5" - Add 5 to the record, record is now [5]. "2" - Add 2 to the record, record is now [5, 2]. "C" - Invalidate and remove the previous score, record is now [5]. "D" - Add 2 * 5 = 10 to the record, record is now [5, 10]. "+" - Add 5 + 10 = 15 to the record, record is now [5, 10, 15]. The total sum is 5 + 10 + 15 = 30.

Example 2: Input: ops = ["5","-2","4","C","D","9","+","+"] Output: 27 Explanation: "5" - Add 5 to the record, record is now [5]. "-2" - Add -2 to the record, record is now [5, -2]. "4" - Add 4 to the record, record is now [5, -2, 4]. "C" - Invalidate and remove the previous score, record is now [5, -2]. "D" - Add 2 * -2 = -4 to the record, record is now [5, -2, -4]. "9" - Add 9 to the record, record is now [5, -2, -4, 9]. "+" - Add -4 + 9 = 5 to the record, record is now [5, -2, -4, 9, 5]. "+" - Add 9 + 5 = 14 to the record, record is now [5, -2, -4, 9, 5, 14]. The total sum is 5 + -2 + -4 + 9 + 5 + 14 = 27.

Example 3: Input: ops = ["1","C"] Output: 0 Explanation: "1" - Add 1 to the record, record is now [1]. "C" - Invalidate and remove the previous score, record is now []. Since the record is empty, the total sum is 0.

Constraints:

1 <= operations.length <= 1000 operations[i] is "C", "D", "+", or a string representing an integer in the range [-3 * 104, 3 * 104]. For operation "+", there will always be at least two previous scores on the record. For operations "C" and "D", there will always be at least one previous score on the record.

==
Problem Number: 133 URL: https://leetcode.com/problems/binary-number-with-alternating-bits Title: 693. Binary Number with Alternating Bits Problem Description: Given a positive integer, check whether it has alternating bits: namely, if two adjacent bits will always have different values. Example 1: Input: n = 5 Output: true Explanation: The binary representation of 5 is: 101

Example 2: Input: n = 7 Output: false Explanation: The binary representation of 7 is: 111. Example 3: Input: n = 11 Output: false Explanation: The binary representation of 11 is: 1011. Constraints:

1 <= n <= 231 - 1

==
Problem Number: 134 URL: https://leetcode.com/problems/count-binary-substrings Title: 696. Count Binary Substrings Problem Description: Given a binary string s, return the number of non-empty substrings that have the same number of 0's and 1's, and all the 0's and all the 1's in these substrings are grouped consecutively. Substrings that occur multiple times are counted the number of times they occur. Example 1: Input: s = "00110011" Output: 6 Explanation: There are 6 substrings that have equal number of consecutive 1's and 0's: "0011", "01", "1100", "10", "0011", and "01". Notice that some of these substrings repeat and are counted the number of times they occur. Also, "00110011" is not a valid substring because all the 0's (and 1's) are not grouped together.

Example 2: Input: s = "10101" Output: 4 Explanation: There are 4 substrings: "10", "01", "10", "01" that have equal number of consecutive 1's and 0's.

Constraints:

1 <= s.length <= 105 s[i] is either '0' or '1'.

==
Problem Number: 135 URL: https://leetcode.com/problems/degree-of-an-array Title: 697. Degree of an Array Problem Description: Given a non-empty array of non-negative integers nums, the degree of this array is defined as the maximum frequency of any one of its elements. Your task is to find the smallest possible length of a (contiguous) subarray of nums, that has the same degree as nums. Example 1: Input: nums = [1,2,2,3,1] Output: 2 Explanation: The input array has a degree of 2 because both elements 1 and 2 appear twice. Of the subarrays that have the same degree: [1, 2, 2, 3, 1], [1, 2, 2, 3], [2, 2, 3, 1], [1, 2, 2], [2, 2, 3], [2, 2] The shortest length is 2. So return 2.

Example 2: Input: nums = [1,2,2,3,1,4,2] Output: 6 Explanation: The degree is 3 because the element 2 is repeated 3 times. So [2,2,3,1,4,2] is the shortest subarray, therefore returning 6.

Constraints:

nums.length will be between 1 and 50,000. nums[i] will be an integer between 0 and 49,999.

==
Problem Number: 136 URL: https://leetcode.com/problems/search-in-a-binary-search-tree Title: 700. Search in a Binary Search Tree Problem Description: You are given the root of a binary search tree (BST) and an integer val. Find the node in the BST that the node's value equals val and return the subtree rooted with that node. If such a node does not exist, return null. Example 1:

Input: root = [4,2,7,1,3], val = 2 Output: [2,1,3]

Example 2:

Input: root = [4,2,7,1,3], val = 5 Output: []

Constraints:

The number of nodes in the tree is in the range [1, 5000]. 1 <= Node.val <= 107 root is a binary search tree. 1 <= val <= 107

==
Problem Number: 137 URL: https://leetcode.com/problems/kth-largest-element-in-a-stream Title: 703. Kth Largest Element in a Stream Problem Description: Design a class to find the kth largest element in a stream. Note that it is the kth largest element in the sorted order, not the kth distinct element. Implement KthLargest class:

KthLargest(int k, int[] nums) Initializes the object with the integer k and the stream of integers nums. int add(int val) Appends the integer val to the stream and returns the element representing the kth largest element in the stream.

Example 1: Input ["KthLargest", "add", "add", "add", "add", "add"] [[3, [4, 5, 8, 2]], [3], [5], [10], [9], [4]] Output [null, 4, 5, 5, 8, 8]

Explanation KthLargest kthLargest = new KthLargest(3, [4, 5, 8, 2]); kthLargest.add(3); // return 4 kthLargest.add(5); // return 5 kthLargest.add(10); // return 5 kthLargest.add(9); // return 8 kthLargest.add(4); // return 8

Constraints:

1 <= k <= 104 0 <= nums.length <= 104 -104 <= nums[i] <= 104 -104 <= val <= 104 At most 104 calls will be made to add. It is guaranteed that there will be at least k elements in the array when you search for the kth element.

==
Problem Number: 138 URL: https://leetcode.com/problems/binary-search

Title: 704. Binary Search Problem Description: Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1. You must write an algorithm with O(log n) runtime complexity. Example 1: Input: nums = [-1,0,3,5,9,12], target = 9 Output: 4 Explanation: 9 exists in nums and its index is 4

Example 2: Input: nums = [-1,0,3,5,9,12], target = 2 Output: -1 Explanation: 2 does not exist in nums so return -1

Constraints:

1 <= nums.length <= 104 -104 < nums[i], target < 104 All the integers in nums are unique. nums is sorted in ascending order.

==
Problem Number: 139 URL: https://leetcode.com/problems/design-hashset Title: 705. Design HashSet Problem Description: Design a HashSet without using any built-in hash table libraries. Implement MyHashSet class:

void add(key) Inserts the value key into the HashSet. bool contains(key) Returns whether the value key exists in the HashSet or not. void remove(key) Removes the value key in the HashSet. If key does not exist in the HashSet, do nothing.

Example 1: Input ["MyHashSet", "add", "add", "contains", "contains", "add", "contains", "remove", "contains"] [[], [1], [2], [1], [3], [2], [2], [2], [2]] Output [null, null, null, true, false, null, true, null, false]

Explanation MyHashSet myHashSet = new MyHashSet(); myHashSet.add(1); // set = [1] myHashSet.add(2); // set = [1, 2] myHashSet.contains(1); // return True myHashSet.contains(3); // return False, (not found) myHashSet.add(2); // set = [1, 2] myHashSet.contains(2); // return True myHashSet.remove(2); // set = [1] myHashSet.contains(2); // return False, (already removed) Constraints:

0 <= key <= 106 At most 104 calls will be made to add, remove, and contains.

==
Problem Number: 140 URL: https://leetcode.com/problems/design-hashmap Title: 706. Design HashMap Problem Description: Design a HashMap without using any built-in hash table libraries. Implement the MyHashMap class:

MyHashMap() initializes the object with an empty map. void put(int key, int value) inserts a (key, value) pair into the HashMap. If the key already exists in the map, update the corresponding value. int get(int key) returns the value to which the specified key is mapped, or -1 if this map contains no mapping for the key. void remove(key) removes the key and its corresponding value if the map contains the mapping for the key.

Example 1: Input ["MyHashMap", "put", "put", "get", "get", "put", "get", "remove", "get"] [[], [1, 1], [2, 2], [1], [3], [2, 1], [2], [2], [2]] Output [null, null, null, 1, -1, null, 1, null, -1]

Explanation MyHashMap myHashMap = new MyHashMap(); myHashMap.put(1, 1); // The map is now [[1,1]] myHashMap.put(2, 2); // The map is now [[1,1], [2,2]] myHashMap.get(1); // return 1, The map is now [[1,1], [2,2]] myHashMap.get(3); // return -1 (i.e., not found), The map is now [[1,1], [2,2]] myHashMap.put(2, 1); // The map is now [[1,1], [2,1]] (i.e., update the existing value) myHashMap.get(2); // return 1, The map is now [[1,1], [2,1]] myHashMap.remove(2); // remove the mapping for 2, The map is now [[1,1]] myHashMap.get(2); // return -1 (i.e., not found), The map is now [[1,1]]

Constraints:

0 <= key, value <= 106 At most 104 calls will be made to put, get, and remove.

==

Problem Number: 141 URL: https://leetcode.com/problems/to-lower-case Title: 709. To Lower Case Problem Description: Given a string s, return the string after replacing every uppercase letter with the same lowercase letter. Example 1: Input: s = "Hello" Output: "hello"

Example 2: Input: s = "here" Output: "here"

Example 3: Input: s = "LOVELY" Output: "lovely"

Constraints:

1 <= s.length <= 100 s consists of printable ASCII characters.

==

Problem Number: 142 URL: https://leetcode.com/problems/1-bit-and-2-bit-characters Title: 717. 1-bit and 2-bit Characters Problem Description: We have two special characters:

The first character can be represented by one bit 0. The second character can be represented by two bits (10 or 11).

Given a binary array bits that ends with 0, return true if the last character must be a one-bit character. Example 1: Input: bits = [1,0,0] Output: true Explanation: The only way to decode it is two-bit character and one-bit character. So the last character is one-bit character.

Example 2: Input: bits = [1,1,1,0] Output: false Explanation: The only way to decode it is two-bit character and two-bit character. So the last character is not one-bit character.

Constraints:

1 <= bits.length <= 1000 bits[i] is either 0 or 1.

==

Problem Number: 143 URL: https://leetcode.com/problems/find-pivot-index Title: 724. Find Pivot Index Problem Description: Given an array of integers nums, calculate the pivot index of this array. The pivot index is the index where the sum of all the numbers strictly to the left of the index is equal to the sum of all the numbers strictly to the index's right. If the index is on the left edge of the array, then the left sum is 0 because there are no elements to the left. This also applies to the right edge of the array. Return the leftmost pivot index. If no such index exists, return -1.   Example 1: Input: nums = [1,7,3,6,5,6] Output: 3 Explanation: The pivot index is 3. Left sum = nums[0] + nums[1] + nums[2] = 1 + 7 + 3 = 11 Right sum = nums[4] + nums[5] = 5 + 6 = 11

Example 2: Input: nums = [1,2,3] Output: -1 Explanation: There is no index that satisfies the conditions in the problem statement. Example 3: Input: nums = [2,1,-1] Output: 0 Explanation: The pivot index is 0. Left sum = 0 (no elements to the left of index 0) Right sum = nums[1] + nums[2] = 1 + -1 = 0

Constraints:

1 <= nums.length <= 104 -1000 <= nums[i] <= 1000

Note: This question is the same as 1991: https://leetcode.com/problems/find-the-middle-index-in-array/

==
Problem Number: 144 URL: https://leetcode.com/problems/self-dividing-numbers Title: 728. Self Dividing Numbers Problem Description: A self-dividing number is a number that is divisible by every digit it contains.

For example, 128 is a self-dividing number because 128 % 1 == 0, 128 % 2 == 0, and 128 % 8 == 0.

A self-dividing number is not allowed to contain the digit zero. Given two integers left and right, return a list of all the self-dividing numbers in the range [left, right].   Example 1: Input: left = 1, right = 22 Output: [1,2,3,4,5,6,7,8,9,11,12,15,22] Example 2: Input: left = 47, right = 85 Output: [48,55,66,77]

Constraints:

1 <= left <= right <= 104

==
Problem Number: 145 URL: https://leetcode.com/problems/flood-fill Title: 733. Flood Fill Problem Description: An image is represented by an m x n integer grid image where image[i][j] represents the pixel value of the image. You are also given three integers sr, sc, and color. You should perform a flood fill on the image starting from the pixel image[sr][sc]. To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color. Return the modified image after performing the flood fill.   Example 1:

Input: image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2 Output:

[[2,2,2],[2,2,0],[2,0,1]] Explanation: From the center of the image with position (sr, sc) = (1, 1) (i.e., the red pixel), all pixels connected by a path of the same color as the starting pixel (i.e., the blue pixels) are colored with the new color. Note the bottom corner is not colored 2, because it is not 4-directionally connected to the starting pixel.

Example 2: Input: image = [[0,0,0],[0,0,0]], sr = 0, sc = 0, color = 0 Output: [[0,0,0],[0,0,0]] Explanation: The starting pixel is already colored 0, so no changes are made to the image.

Constraints:

m == image.length n == image[i].length 1 <= m, n <= 50 0 <= image[i][j], color < 216 0 <= sr < m 0 <= sc < n

==
Problem Number: 146 URL: https://leetcode.com/problems/find-smallest-letter-greater-than-target Title: 744. Find Smallest Letter Greater Than Target Problem Description: You are given an array of characters letters that is sorted in non-decreasing order, and a character target. There are at least two different characters in letters. Return the smallest character in letters that is lexicographically greater than target. If such a character does not exist, return the first character in letters. Example 1: Input: letters = ["c","f","j"], target = "a" Output: "c" Explanation: The smallest character that is lexicographically greater than 'a' in letters is 'c'.

Example 2: Input: letters = ["c","f","j"], target = "c" Output: "f" Explanation: The smallest character that is lexicographically greater than 'c' in letters is 'f'.

Example 3: Input: letters = ["x","x","y","y"], target = "z" Output: "x" Explanation: There are no characters in letters that is lexicographically greater than 'z' so we return letters[0].

Constraints:

2 <= letters.length <= 104 letters[i] is a lowercase English letter. letters is sorted in non-decreasing order. letters contains at least two different characters. target is a lowercase English letter.

==
Problem Number: 147 URL: https://leetcode.com/problems/min-cost-climbing-stairs Title: 746. Min Cost Climbing Stairs Problem Description: You are given an integer array cost where cost[i] is the cost of ith step on a staircase. Once you pay the cost, you can either climb one or two steps. You can either start from the step with index 0, or the step with index 1. Return the minimum cost to reach the top of the floor. Example 1: Input: cost = [10,15,20] Output: 15 Explanation: You will start at index 1. - Pay 15 and climb two steps to reach the top. The total cost is 15.

Example 2: Input: cost = [1,100,1,1,1,100,1,1,100,1] Output: 6 Explanation: You will start at index 0. - Pay 1 and climb two steps to reach index 2. - Pay

1 and climb two steps to reach index 4. - Pay 1 and climb two steps to reach index 6. - Pay 1 and climb one step to reach index 7. - Pay 1 and climb two steps to reach index 9. - Pay 1 and climb one step to reach the top. The total cost is 6.

Constraints:

2 <= cost.length <= 1000 0 <= cost[i] <= 999

==
Problem Number: 148 URL: https://leetcode.com/problems/largest-number-at-least-twice-of-others Title: 747. Largest Number At Least Twice of Others Problem Description: You are given an integer array nums where the largest integer is unique. Determine whether the largest element in the array is at least twice as much as every other number in the array. If it is, return the index of the largest element, or return -1 otherwise. Example 1: Input: nums = [3,6,1,0] Output: 1 Explanation: 6 is the largest integer. For every other number in the array x, 6 is at least twice as big as x. The index of value 6 is 1, so we return 1.

Example 2: Input: nums = [1,2,3,4] Output: -1 Explanation: 4 is less than twice the value of 3, so we return -1.

Constraints:

2 <= nums.length <= 50 0 <= nums[i] <= 100 The largest element in nums is unique.

==
Problem Number: 149 URL: https://leetcode.com/problems/shortest-completing-word Title: 748. Shortest Completing Word Problem Description: Given a string licensePlate and an array of strings words, find the shortest completing word in words. A completing word is a word that contains all the letters in licensePlate. Ignore numbers and spaces in licensePlate, and treat letters as case insensitive. If a letter appears more than once in licensePlate, then it must appear in the word the same number of times or more. For example, if licensePlate = "aBc 12c", then it contains letters 'a', 'b' (ignoring case), and 'c' twice. Possible completing words are "abccdef", "caaacab", and "cbca". Return the shortest completing word in words. It is guaranteed an answer exists. If there are multiple shortest completing words, return the first one that occurs in words. Example 1: Input: licensePlate = "1s3 PSt", words = ["step","steps","stripe","stepple"] Output: "steps" Explanation: licensePlate contains letters 's', 'p', 's' (ignoring case), and 't'. "step" contains 't' and 'p', but only contains 1 's'. "steps" contains 't', 'p', and both 's' characters. "stripe" is missing an 's'. "stepple" is missing an 's'. Since "steps" is the only word containing all the letters, that is the answer.

Example 2: Input: licensePlate = "1s3 456", words = ["looks","pest","stew","show"] Output: "pest" Explanation: licensePlate only contains the letter 's'. All the words contain 's', but among these "pest", "stew", and "show" are shortest. The answer is "pest" because it is the word that appears earliest of the 3.

Constraints:

1 <= licensePlate.length <= 7 licensePlate contains digits, letters (uppercase or lowercase), or space ' '. 1 <= words.length <= 1000 1 <= words[i].length <= 15 words[i] consists of lower case English letters.

==

Problem Number: 150 URL: https://leetcode.com/problems/prime-number-of-set-bits-in-binary-representation Title: 762. Prime Number of Set Bits in Binary Representation Problem Description: Given two integers left and right, return the count of numbers in the inclusive range [left, right] having a prime number of set bits in their binary representation. Recall that the number of set bits an integer has is the number of 1's present when written in binary.

For example, 21 written in binary is 10101, which has 3 set bits.

Example 1: Input: left = 6, right = 10 Output: 4 Explanation: 6 -> 110 (2 set bits, 2 is prime) 7 -> 111 (3 set bits, 3 is prime) 8 -> 1000 (1 set bit, 1 is not prime) 9 -> 1001 (2 set bits, 2 is prime) 10 -> 1010 (2 set bits, 2 is prime) 4 numbers have a prime number of set bits.

Example 2: Input: left = 10, right = 15 Output: 5 Explanation: 10 -> 1010 (2 set bits, 2 is prime) 11 -> 1011 (3 set bits, 3 is prime) 12 -> 1100 (2 set bits, 2 is prime) 13 -> 1101 (3 set bits, 3 is prime) 14 -> 1110 (3 set bits, 3 is prime) 15 -> 1111 (4 set bits, 4 is not prime) 5 numbers have a prime number of set bits.

Constraints:

1 <= left <= right <= 106 0 <= right - left <= 104

==

Problem Number: 151 URL: https://leetcode.com/problems/toeplitz-matrix Title: 766. Toeplitz Matrix Problem Description: Given an m x n matrix, return true if the matrix is Toeplitz. Otherwise, return false. A matrix is Toeplitz if every diagonal from top-left to bottom-right has the same elements. Example 1:

Input: matrix = [[1,2,3,4],[5,1,2,3],[9,5,1,2]] Output: true Explanation: In the above grid, the diagonals are: "[9]", "[5, 5]", "[1, 1, 1]", "[2, 2, 2]", "[3, 3]", "[4]". In each diagonal all elements are the same, so the answer is True.

Example 2:

Input: matrix = [[1,2],[2,2]] Output: false Explanation: The diagonal "[1, 2]" has different elements.

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 20 0 <= matrix[i][j] <= 99

Follow up:

What if the matrix is stored on disk, and the memory is limited such that you can only load at most one row of the matrix into the memory at once? What if the matrix is so large that you can only load up a partial row into the memory at once?

==
Problem Number: 152 URL: https://leetcode.com/problems/jewels-and-stones Title: 771. Jewels and Stones Problem Description: You're given strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type of stone you have. You want to know how many of the stones you have are also jewels. Letters are case sensitive, so "a" is considered a different type of stone from "A". Example 1: Input: jewels = "aA", stones = "aAAbbbb" Output: 3 Example 2: Input: jewels = "z", stones = "ZZ" Output: 0

Constraints:

1 <= jewels.length, stones.length <= 50 jewels and stones consist of only English letters. All the characters of jewels are unique.

==
Problem Number: 153 URL: https://leetcode.com/problems/minimum-distance-between-bst-nodes Title: 783. Minimum Distance Between BST Nodes Problem Description: Given the root of a Binary Search Tree (BST), return the minimum difference between the values of any two different nodes in the tree. Example 1:

Input: root = [4,2,6,1,3] Output: 1

Example 2:

Input: root = [1,0,48,null,null,12,49] Output: 1

Constraints:

The number of nodes in the tree is in the range [2, 100]. 0 <= Node.val <= 105

Note: This question is the same as 530: https://leetcode.com/problems/minimum-absolute-difference-in-bst/

==
Problem Number: 154 URL: https://leetcode.com/problems/rotate-string Title: 796. Rotate String Problem Description: Given two strings s and goal, return true if and only if s can become goal after some number of shifts on s. A shift on s consists of moving the leftmost character of s to the rightmost position.

For example, if s = ”abcde”, then it will be ”bcdea” after one shift.

Example 1: Input: s = ”abcde”, goal = ”cdeab” Output: true Example 2: Input: s = ”abcde”, goal = ”abced” Output: false

Constraints:

1 <= s.length, goal.length <= 100 s and goal consist of lowercase English letters.

==

Problem Number: 155 URL: https://leetcode.com/problems/unique-morse-code-words Title: 804. Unique Morse Code Words Problem Description: International Morse Code defines a standard encoding where each letter is mapped to a series of dots and dashes, as follows:

’a’ maps to ”.-”, ’b’ maps to ”-...”, ’c’ maps to ”-.-.”, and so on.

For convenience, the full table for the 26 letters of the English alphabet is given below: [”.-”,”-...”,”-.-.”,”-..”,”.”,”..-.”,”–.”,”....”,”..”,”.—”,”-.-”,”.-..”,”–”,”-.”,”—”,”.–.”,”–.-”,”.-.”,”...”,”-”,”..-”,”...-”,”.–”,”-..-”,”-.–”,”–..”] Given an array of strings words where each word can be written as a concatenation of the Morse code of each letter.

For example, ”cab” can be written as ”-.-..–...”, which is the concatenation of ”-.-.”, ”.-”, and ”-...”. We will call such a concatenation the transformation of a word.

Return the number of different transformations among all words we have. Example 1: Input: words = [”gin”,”zen”,”gig”,”msg”] Output: 2 Explanation: The transformation of each word is: ”gin” -> ”–...-.” ”zen” -> ”–...-.” ”gig” -> ”–...–.” ”msg” -> ”–...–.” There are 2 different transformations: ”–...-.” and ”–...–.”.

Example 2: Input: words = [”a”] Output: 1

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 12 words[i] consists of lowercase English letters.

==

Problem Number: 156 URL: https://leetcode.com/problems/number-of-lines-to-write-string Title: 806. Number of Lines To Write String Problem Description: You are given a string s of lowercase English letters and an array widths denoting how many pixels wide each lowercase English letter is. Specifically, widths[0] is the width of ’a’, widths[1] is the width of ’b’, and so on. You are trying to write s across several lines, where each line is no longer than 100 pixels. Starting at the beginning of s, write as many letters on the first line such that the total width does not exceed 100 pixels. Then, from where you stopped in s, continue writing as many letters as you can on the second line. Continue this process until you have written all of s. Return an array result of length 2 where:

result[0] is the total number of lines. result[1] is the width of the last line in pixels.

Example 1: Input: widths = [10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10], s = "abcdefghijklmnopqrstuvwxyz" Output: [3,60] Explanation: You can write s as follows: abcdefghij // 100 pixels wide klmnopqrst // 100 pixels wide uvwxyz // 60 pixels wide There are a total of 3 lines, and the last line is 60 pixels wide. Example 2: Input: widths = [4,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10], s = "bbbcccdddaaa" Output: [2,4] Explanation: You can write s as follows: bbbcccdddaa // 98 pixels wide a // 4 pixels wide There are a total of 2 lines, and the last line is 4 pixels wide. Constraints:

widths.length == 26 2 <= widths[i] <= 10 1 <= s.length <= 1000 s contains only lowercase English letters.

== Problem Number: 157 URL: https://leetcode.com/problems/largest-triangle-area Title: 812. Largest Triangle Area Problem Description: Given an array of points on the X-Y plane points where points[i] = [xi, yi], return the area of the largest triangle that can be formed by any three different points. Answers within 10-5 of the actual answer will be accepted. Example 1:

Input: points = [[0,0],[0,1],[1,0],[0,2],[2,0]] Output: 2.00000 Explanation: The five points are shown in the above figure. The red triangle is the largest.

Example 2: Input: points = [[1,0],[0,0],[0,1]] Output: 0.50000

Constraints:

3 <= points.length <= 50 -50 <= xi, yi <= 50 All the given points are unique.

== Problem Number: 158 URL: https://leetcode.com/problems/most-common-word Title: 819. Most Common Word Problem Description: Given a string paragraph and a string array of the banned words banned, return the most frequent word that is not banned. It is guaranteed there is at least one word that is not banned, and that the answer is unique. The words in paragraph are case-insensitive and the answer should be returned in lowercase. Example 1: Input: paragraph = "Bob hit a ball, the hit BALL flew far after it was hit.", banned = ["hit"] Output: "ball" Explanation: "hit" occurs 3 times, but it is a banned word. "ball" occurs twice (and no other word does), so it is the most frequent non-banned word in the paragraph. Note that words in the paragraph are not case sensitive, that punctuation is ignored (even if adjacent to words, such as "ball,"), and that "hit" isn't the answer even though it occurs more because it is banned.

Example 2: Input: paragraph = "a.", banned = [] Output: "a"

Constraints:

1 <= paragraph.length <= 1000 paragraph consists of English letters, space ' ', or one of the symbols: "!?',;.". 0 <= banned.length <= 100 1 <= banned[i].length <= 10 banned[i] consists of only lowercase English letters.

==

Problem Number: 159 URL: https://leetcode.com/problems/shortest-distance-to-a-character Title: 821. Shortest Distance to a Character Problem Description: Given a string s and a character c that occurs in s, return an array of integers answer where answer.length == s.length and answer[i] is the distance from index i to the closest occurrence of character c in s. The distance between two indices i and j is abs(i - j), where abs is the absolute value function. Example 1: Input: s = "loveleetcode", c = "e" Output: [3,2,1,0,1,0,0,1,2,2,1,0] Explanation: The character 'e' appears at indices 3, 5, 6, and 11 (0-indexed). The closest occurrence of 'e' for index 0 is at index 3, so the distance is abs(0 - 3) = 3. The closest occurrence of 'e' for index 1 is at index 3, so the distance is abs(1 - 3) = 2. For index 4, there is a tie between the 'e' at index 3 and the 'e' at index 5, but the distance is still the same: abs(4 - 3) == abs(4 - 5) = 1. The closest occurrence of 'e' for index 8 is at index 6, so the distance is abs(8 - 6) = 2.

Example 2: Input: s = "aaab", c = "b" Output: [3,2,1,0]

Constraints:

1 <= s.length <= 104 s[i] and c are lowercase English letters. It is guaranteed that c occurs at least once in s.

==

Problem Number: 160 URL: https://leetcode.com/problems/goat-latin Title: 824. Goat Latin Problem Description: You are given a string sentence that consist of words separated by spaces. Each word consists of lowercase and uppercase letters only. We would like to convert the sentence to "Goat Latin" (a made-up language similar to Pig Latin.) The rules of Goat Latin are as follows:

If a word begins with a vowel ('a', 'e', 'i', 'o', or 'u'), append "ma" to the end of the word.

For example, the word "apple" becomes "applema".

If a word begins with a consonant (i.e., not a vowel), remove the first letter and append it to the end, then add "ma".

For example, the word "goat" becomes "oatgma".

Add one letter 'a' to the end of each word per its word index in the sentence, starting with 1.

For example, the first word gets "a" added to the end, the second word gets "aa" added to the end, and so on.

Return the final sentence representing the conversion from sentence to Goat Latin. Example 1: Input: sentence = "I speak Goat Latin" Output: "Imaa peaksmaaa oatGmaaaa atinLmaaaaa" Example 2: Input: sentence = "The quick brown fox jumped over the lazy dog" Output: "heTmaa uickqmaaa rownbmaaaa oxfmaaaaa umpedjmaaaaaa overmaaaaaaa hetmaaaaaaaa azylmaaaaaaaaa ogdmaaaaaaaaaa"

Constraints:

1 <= sentence.length <= 150 sentence consists of English letters and spaces. sentence has no leading or trailing spaces. All the words in sentence are separated by a single space.

==

Problem Number: 161 URL: https://leetcode.com/problems/positions-of-large-groups Title: 830. Positions of Large Groups Problem Description: In a string s of lowercase letters, these letters form consecutive groups of the same character. For example, a string like s = "abbxxxxzyy" has the groups "a", "bb", "xxxx", "z", and "yy". A group is identified by an interval [start, end], where start and end denote the start and end indices (inclusive) of the group. In the above example, "xxxx" has the interval [3,6]. A group is considered large if it has 3 or more characters. Return the intervals of every large group sorted in increasing order by start index. Example 1: Input: s = "abbxxxxzzy" Output: [[3,6]] Explanation: "xxxx" is the only large group with start index 3 and end index 6.

Example 2: Input: s = "abc" Output: [] Explanation: We have groups "a", "b", and "c", none of which are large groups.

Example 3: Input: s = "abcdddeeeeaabbbcd" Output: [[3,5],[6,9],[12,14]] Explanation: The large groups are "ddd", "eeee", and "bbb".

Constraints:

1 <= s.length <= 1000 s contains lowercase English letters only.

==

Problem Number: 162 URL: https://leetcode.com/problems/flipping-an-image Title: 832. Flipping an Image Problem Description: Given an n x n binary matrix image, flip the image horizontally, then invert it, and return the resulting image. To flip an image horizontally means that each row of the image is reversed.

For example, flipping [1,1,0] horizontally results in [0,1,1].

To invert an image means that each 0 is replaced by 1, and each 1 is replaced by 0.

For example, inverting [0,1,1] results in [1,0,0].

Example 1: Input: image = [[1,1,0],[1,0,1],[0,0,0]] Output: [[1,0,0],[0,1,0],[1,1,1]] Explanation: First reverse each row: [[0,1,1],[1,0,1],[0,0,0]]. Then, invert the image: [[1,0,0],[0,1,0],[1,1,1]]

Example 2: Input: image = [[1,1,0,0],[1,0,0,1],[0,1,1,1],[1,0,1,0]] Output: [[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]] Explanation: First reverse each row: [[0,0,1,1],[1,0,0,1],[1,1,1,0],[0,1,0,1]]. Then invert the image: [[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]

Constraints:

n == image.length n == image[i].length 1 <= n <= 20 images[i][j] is either 0 or 1.

==

Problem Number: 163 URL: https://leetcode.com/problems/rectangle-overlap Title: 836. Rectangle Overlap Problem Description: An axis-aligned rectangle is represented as a list [x1, y1, x2, y2], where (x1, y1) is the coordinate of its bottom-left corner, and (x2, y2) is the coordinate of its top-right corner. Its top and bottom edges are parallel to the X-axis, and its left and right edges are parallel to the Y-axis. Two rectangles overlap if the area of their intersection is positive. To be clear, two rectangles that only touch at the corner or edges do not overlap. Given two axis-aligned rectangles rec1 and rec2, return true if they overlap, otherwise return false. Example 1: Input: rec1 = [0,0,2,2], rec2 = [1,1,3,3] Output: true Example 2: Input: rec1 = [0,0,1,1], rec2 = [1,0,2,1] Output: false Example 3: Input: rec1 = [0,0,1,1], rec2 = [2,2,3,3] Output: false

Constraints:

rec1.length == 4 rec2.length == 4 -109 <= rec1[i], rec2[i] <= 109 rec1 and rec2 represent a valid rectangle with a non-zero area.

==

Problem Number: 164 URL: https://leetcode.com/problems/backspace-string-compare Title: 844. Backspace String Compare Problem Description: Given two strings s and t, return true if they are equal when both are typed into empty text editors. '#' means a backspace character. Note that after backspacing an empty text, the text will continue empty. Example 1: Input: s = "ab#c", t = "ad#c" Output: true Explanation: Both s and t become "ac".

Example 2: Input: s = "ab##", t = "c#d#" Output: true Explanation: Both s and t become "".

Example 3: Input: s = "a#c", t = "b" Output: false Explanation: s becomes "c" while t becomes "b".

Constraints:

1 <= s.length, t.length <= 200 s and t only contain lowercase letters and '#' characters.

Follow up: Can you solve it in O(n) time and O(1) space?

==

Problem Number: 165 URL: https://leetcode.com/problems/buddy-strings

Title: 859. Buddy Strings Problem Description: Given two strings s and goal, return true if you can swap two letters in s so the result is equal to goal, otherwise, return false. Swapping letters is defined as taking two indices i and j (0-indexed) such that i != j and swapping the characters at s[i] and s[j].

For example, swapping at indices 0 and 2 in "abcd" results in "cbad".

Example 1: Input: s = "ab", goal = "ba" Output: true Explanation: You can swap s[0] = 'a' and s[1] = 'b' to get "ba", which is equal to goal.

Example 2: Input: s = "ab", goal = "ab" Output: false Explanation: The only letters you can swap are s[0] = 'a' and s[1] = 'b', which results in "ba" != goal.

Example 3: Input: s = "aa", goal = "aa" Output: true Explanation: You can swap s[0] = 'a' and s[1] = 'a' to get "aa", which is equal to goal.

Constraints:

1 <= s.length, goal.length <= 2 * 104 s and goal consist of lowercase letters.

==
Problem Number: 166 URL: https://leetcode.com/problems/lemonade-change Title: 860. Lemonade Change Problem Description: At a lemonade stand, each lemonade costs $5. Customers are standing in a queue to buy from you and order one at a time (in the order specified by bills). Each customer will only buy one lemonade and pay with either a $5, $10, or $20 bill. You must provide the correct change to each customer so that the net transaction is that the customer pays $5. Note that you do not have any change in hand at first. Given an integer array bills where bills[i] is the bill the ith customer pays, return true if you can provide every customer with the correct change, or false otherwise. Example 1: Input: bills = [5,5,5,10,20] Output: true Explanation: From the first 3 customers, we collect three $5 bills in order. From the fourth customer, we collect a $10 bill and give back a $5. From the fifth customer, we give a $10 bill and a $5 bill. Since all customers got correct change, we output true.

Example 2: Input: bills = [5,5,10,10,20] Output: false Explanation: From the first two customers in order, we collect two $5 bills. For the next two customers in order, we collect a $10 bill and give back a $5 bill. For the last customer, we can not give the change of $15 back because we only have two $10 bills. Since not every customer received the correct change, the answer is false.

Constraints:

1 <= bills.length <= 105 bills[i] is either 5, 10, or 20.

==
Problem Number: 167 URL: https://leetcode.com/problems/transpose-matrix Title: 867. Transpose Matrix Problem Description: Given a 2D integer array matrix, return the transpose of matrix. The transpose of a matrix is the matrix flipped over its main diagonal, switching the matrix's row and column indices.

Example 1: Input: matrix = [[1,2,3],[4,5,6],[7,8,9]] Output: [[1,4,7],[2,5,8],[3,6,9]]

Example 2: Input: matrix = [[1,2,3],[4,5,6]] Output: [[1,4],[2,5],[3,6]]

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 1000 1 <= m * n <= 105 -109 <= matrix[i][j] <= 109

==
Problem Number: 168 URL: https://leetcode.com/problems/binary-gap Title: 868. Binary Gap Problem Description: Given a positive integer n, find and return the longest distance between any two adjacent 1's in the binary representation of n. If there are no two adjacent 1's, return 0. Two 1's are adjacent if there are only 0's separating them (possibly no 0's). The distance between two 1's is the absolute difference between their bit positions. For example, the two 1's in "1001" have a distance of 3. Example 1: Input: n = 22 Output: 2 Explanation: 22 in binary is "10110". The first adjacent pair of 1's is "10110" with a distance of 2. The second adjacent pair of 1's is "10110" with a distance of 1. The answer is the largest of these two distances, which is 2. Note that "10110" is not a valid pair since there is a 1 separating the two 1's underlined.

Example 2: Input: n = 8 Output: 0 Explanation: 8 in binary is "1000". There are not any adjacent pairs of 1's in the binary representation of 8, so we return 0.

Example 3: Input: n = 5 Output: 2 Explanation: 5 in binary is "101".

Constraints:

1 <= n <= 109

==
Problem Number: 169 URL: https://leetcode.com/problems/leaf-similar-trees Title: 872. Leaf-Similar Trees Problem Description: Consider all the leaves of a binary tree, from left to right order, the values of those leaves form a leaf value sequence.

For example, in the given tree above, the leaf value sequence is (6, 7, 4, 9, 8). Two binary trees are considered leaf-similar if their leaf value sequence is the same. Return true if and only if the two given trees with head nodes root1 and root2 are leaf-similar. Example 1:

Input: root1 = [3,5,1,6,2,9,8,null,null,7,4], root2 = [3,5,1,6,7,4,2,null,null,null,null,null,null,9,8] Output: true

Example 2:

Input: root1 = [1,2,3], root2 = [1,3,2] Output: false

Constraints:

The number of nodes in each tree will be in the range [1, 200]. Both of the given trees will have values in the range [0, 200].

===
Problem Number: 170 URL: https://leetcode.com/problems/middle-of-the-linked-list Title: 876. Middle of the Linked List Problem Description: Given the head of a singly linked list, return the middle node of the linked list. If there are two middle nodes, return the second middle node. Example 1:

Input: head = [1,2,3,4,5] Output: [3,4,5] Explanation: The middle node of the list is node 3.

Example 2:

Input: head = [1,2,3,4,5,6] Output: [4,5,6] Explanation: Since the list has two middle nodes with values 3 and 4, we return the second one.

Constraints:

The number of nodes in the list is in the range [1, 100]. 1 <= Node.val <= 100

===
Problem Number: 171 URL: https://leetcode.com/problems/projection-area-of-3d-shapes Title: 883. Projection Area of 3D Shapes Problem Description: You are given an n x n grid where we place some 1 x 1 x 1 cubes that are axis-aligned with the x, y, and z axes. Each value v = grid[i][j] represents a tower of v cubes placed on top of the cell (i, j). We view the projection of these cubes onto the xy, yz, and zx planes. A projection is like a shadow, that maps our 3-dimensional figure to a 2-dimensional plane. We are viewing the "shadow" when looking at the cubes from the top, the front, and the side. Return the total area of all three projections. Example 1:

Input: grid = [[1,2],[3,4]] Output: 17 Explanation: Here are the three projections ("shadows") of the shape made with each axis-aligned plane.

Example 2: Input: grid = [[2]] Output: 5

Example 3: Input: grid = [[1,0],[0,2]] Output: 8

Constraints:

n == grid.length == grid[i].length 1 <= n <= 50 0 <= grid[i][j] <= 50

===
Problem Number: 172 URL: https://leetcode.com/problems/uncommon-words-from-two-sentences Title: 884. Uncommon Words from Two Sentences Problem Description: A sentence is a string of single-space separated words where each word consists only of lowercase letters. A word is uncommon if it appears exactly once in one of the sentences, and does not appear in the other sentence. Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order. Example 1: Input: s1 = "this apple is

sweet", s2 = "this apple is sour" Output: ["sweet","sour"] Example 2: Input: s1 = "apple apple", s2 = "banana" Output: ["banana"]

Constraints:

1 <= s1.length, s2.length <= 200 s1 and s2 consist of lowercase English letters and spaces. s1 and s2 do not have leading or trailing spaces. All the words in s1 and s2 are separated by a single space.

==

Problem Number: 173 URL: https://leetcode.com/problems/fair-candy-swap Title: 888. Fair Candy Swap Problem Description: Alice and Bob have a different total number of candies. You are given two integer arrays aliceSizes and bobSizes where aliceSizes[i] is the number of candies of the ith box of candy that Alice has and bobSizes[j] is the number of candies of the jth box of candy that Bob has. Since they are friends, they would like to exchange one candy box each so that after the exchange, they both have the same total amount of candy. The total amount of candy a person has is the sum of the number of candies in each box they have. Return an integer array answer where answer[0] is the number of candies in the box that Alice must exchange, and answer[1] is the number of candies in the box that Bob must exchange. If there are multiple answers, you may return any one of them. It is guaranteed that at least one answer exists. Example 1: Input: aliceSizes = [1,1], bobSizes = [2,2] Output: [1,2]

Example 2: Input: aliceSizes = [1,2], bobSizes = [2,3] Output: [1,2]

Example 3: Input: aliceSizes = [2], bobSizes = [1,3] Output: [2,3]

Constraints:

1 <= aliceSizes.length, bobSizes.length <= 104 1 <= aliceSizes[i], bobSizes[j] <= 105 Alice and Bob have a different total number of candies. There will be at least one valid answer for the given input.

==

Problem Number: 174 URL: https://leetcode.com/problems/surface-area-of-3d-shapes Title: 892. Surface Area of 3D Shapes Problem Description: You are given an n x n grid where you have placed some 1 x 1 x 1 cubes. Each value v = grid[i][j] represents a tower of v cubes placed on top of cell (i, j). After placing these cubes, you have decided to glue any directly adjacent cubes to each other, forming several irregular 3D shapes. Return the total surface area of the resulting shapes. Note: The bottom face of each shape counts toward its surface area. Example 1:

Input: grid = [[1,2],[3,4]] Output: 34

Example 2:

Input: grid = [[1,1,1],[1,0,1],[1,1,1]] Output: 32

Example 3:

Input: grid = [[2,2,2],[2,1,2],[2,2,2]] Output: 46

Constraints:

n == grid.length == grid[i].length 1 <= n <= 50 0 <= grid[i][j] <= 50

==
Problem Number: 175 URL: https://leetcode.com/problems/monotonic-array Title: 896. Monotonic Array Problem Description: An array is monotonic if it is either monotone increasing or monotone decreasing. An array nums is monotone increasing if for all i <= j, nums[i] <= nums[j]. An array nums is monotone decreasing if for all i <= j, nums[i] >= nums[j]. Given an integer array nums, return true if the given array is monotonic, or false otherwise. Example 1: Input: nums = [1,2,2,3] Output: true

Example 2: Input: nums = [6,5,4,4] Output: true

Example 3: Input: nums = [1,3,2] Output: false

Constraints:

1 <= nums.length <= 105 -105 <= nums[i] <= 105

==
Problem Number: 176 URL: https://leetcode.com/problems/increasing-order-search-tree Title: 897. Increasing Order Search Tree Problem Description: Given the root of a binary search tree, rearrange the tree in in-order so that the leftmost node in the tree is now the root of the tree, and every node has no left child and only one right child. Example 1:

Input: root = [5,3,6,2,4,null,8,1,null,null,null,7,9] Output: [1,null,2,null,3,null,4,null,5,null,6,null,7,null,8,null,9]

Example 2:

Input: root = [5,1,7] Output: [1,null,5,null,7]

Constraints:

The number of nodes in the given tree will be in the range [1, 100]. 0 <= Node.val <= 1000

==
Problem Number: 177 URL: https://leetcode.com/problems/sort-array-by-parity Title: 905. Sort Array By Parity Problem Description: Given an integer array nums, move all the even integers at the beginning of the array followed by all the odd integers. Return any array that satisfies this condition. Example 1: Input: nums = [3,1,2,4] Output: [2,4,3,1] Explanation: The outputs [4,2,3,1], [2,4,1,3], and [4,2,1,3] would also be accepted.

Example 2: Input: nums = [0] Output: [0]

Constraints:

1 <= nums.length <= 5000 0 <= nums[i] <= 5000

==
Problem Number: 178 URL: https://leetcode.com/problems/smallest-range-i Title: 908. Smallest Range I Problem Description: You are given an integer array nums and an integer k. In one operation, you can choose any index i where 0 <= i < nums.length and change nums[i] to nums[i] + x where x is an integer from the range [-k, k]. You can apply this operation at most once for each index i. The score of nums is the difference between the maximum and minimum elements in nums. Return the minimum score of nums after applying the mentioned operation at most once for each index in it. Example 1: Input: nums = [1], k = 0 Output: 0 Explanation: The score is max(nums) - min(nums) = 1 - 1 = 0.

Example 2: Input: nums = [0,10], k = 2 Output: 6 Explanation: Change nums to be [2, 8]. The score is max(nums) - min(nums) = 8 - 2 = 6.

Example 3: Input: nums = [1,3,6], k = 3 Output: 0 Explanation: Change nums to be [4, 4, 4]. The score is max(nums) - min(nums) = 4 - 4 = 0.

Constraints:

1 <= nums.length <= 104 0 <= nums[i] <= 104 0 <= k <= 104

==
Problem Number: 179 URL: https://leetcode.com/problems/x-of-a-kind-in-a-deck-of-cards Title: 914. X of a Kind in a Deck of Cards Problem Description: You are given an integer array deck where deck[i] represents the number written on the ith card. Partition the cards into one or more groups such that:

Each group has exactly x cards where x > 1, and All the cards in one group have the same integer written on them.

Return true if such partition is possible, or false otherwise. Example 1: Input: deck = [1,2,3,4,4,3,2,1] Output: true Explanation: Possible partition [1,1],[2,2],[3,3],[4,4].

Example 2: Input: deck = [1,1,1,2,2,2,3,3] Output: false Explanation: No possible partition.

Constraints:

1 <= deck.length <= 104 0 <= deck[i] < 104

==
Problem Number: 180 URL: https://leetcode.com/problems/reverse-only-letters Title: 917. Reverse Only Letters Problem Description: Given a string s, reverse the string according to the following rules:

All the characters that are not English letters remain in the same position. All the English letters (lowercase or uppercase) should be reversed.

Return s after reversing it. Example 1: Input: s = "ab-cd" Output: "dc-ba" Example 2: Input: s = "a-bC-dEf-ghIj" Output: "j-Ih-gfE-dCba" Example 3:

Input: s = ”Test1ng-Leet=code-Q!” Output: ”Qedo1ct-eeLg=ntse-T!”

Constraints:

1 <= s.length <= 100 s consists of characters with ASCII values in the range [33, 122]. s does not contain '\"' or '\\'.

==
Problem Number: 181 URL: https://leetcode.com/problems/sort-array-by-parity-ii Title: 922. Sort Array By Parity II Problem Description: Given an array of integers nums, half of the integers in nums are odd, and the other half are even. Sort the array so that whenever nums[i] is odd, i is odd, and whenever nums[i] is even, i is even. Return any answer array that satisfies this condition. Example 1: Input: nums = [4,2,5,7] Output: [4,5,2,7] Explanation: [4,7,2,5], [2,5,4,7], [2,7,4,5] would also have been accepted.

Example 2: Input: nums = [2,3] Output: [2,3]

Constraints:

2 <= nums.length <= 2 * 104 nums.length is even. Half of the integers in nums are even. 0 <= nums[i] <= 1000

Follow Up: Could you solve it in-place?

==
Problem Number: 182 URL: https://leetcode.com/problems/long-pressed-name Title: 925. Long Pressed Name Problem Description: Your friend is typing his name into a keyboard. Sometimes, when typing a character c, the key might get long pressed, and the character will be typed 1 or more times. You examine the typed characters of the keyboard. Return True if it is possible that it was your friends name, with some characters (possibly none) being long pressed. Example 1: Input: name = ”alex”, typed = ”aaleex” Output: true Explanation: 'a' and 'e' in 'alex' were long pressed.

Example 2: Input: name = ”saeed”, typed = ”ssaaedd” Output: false Explanation: 'e' must have been pressed twice, but it was not in the typed output.

Constraints:

1 <= name.length, typed.length <= 1000 name and typed consist of only lowercase English letters.

==
Problem Number: 183 URL: https://leetcode.com/problems/unique-email-addresses Title: 929. Unique Email Addresses Problem Description: Every valid email consists of a local name and a domain name, separated by the '@' sign. Besides lowercase letters, the email may contain one or more '.' or '+'.

For example, in ”alice@leetcode.com”, ”alice” is the local name, and ”leetcode.com” is the domain name.

If you add periods '.' between some characters in the local name part of an email address, mail sent there will be forwarded to the same address without dots in the local name. Note that this rule does not apply to domain names.

For example, "alice.z@leetcode.com" and "alicez@leetcode.com" forward to the same email address.

If you add a plus '+' in the local name, everything after the first plus sign will be ignored. This allows certain emails to be filtered. Note that this rule does not apply to domain names.

For example, "m.y+name@email.com" will be forwarded to "my@email.com".

It is possible to use both of these rules at the same time. Given an array of strings emails where we send one email to each emails[i], return the number of different addresses that actually receive mails. Example 1: Input: emails = ["test.email+alex@leetcode.com","test.e.mail+bob.cathy@leetcode.com","testemail+david@lee.tcode.com"] Output: 2 Explanation: "testemail@leetcode.com" and "testemail@lee.tcode.com" actually receive mails.

Example 2: Input: emails = ["a@leetcode.com","b@leetcode.com","c@leetcode.com"] Output: 3

Constraints:

1 <= emails.length <= 100 1 <= emails[i].length <= 100 emails[i] consist of lowercase English letters, '+', '.' and '@'. Each emails[i] contains exactly one '@' character. All local and domain names are non-empty. Local names do not start with a '+' character. Domain names end with the ".com" suffix.

==

Problem Number: 184 URL: https://leetcode.com/problems/number-of-recent-calls Title: 933. Number of Recent Calls Problem Description: You have a RecentCounter class which counts the number of recent requests within a certain time frame. Implement the RecentCounter class:

RecentCounter() Initializes the counter with zero recent requests. int ping(int t) Adds a new request at time t, where t represents some time in milliseconds, and returns the number of requests that has happened in the past 3000 milliseconds (including the new request). Specifically, return the number of requests that have happened in the inclusive range [t - 3000, t].

It is guaranteed that every call to ping uses a strictly larger value of t than the previous call. Example 1: Input ["RecentCounter", "ping", "ping", "ping", "ping"] [[], [1], [100], [3001], [3002]] Output [null, 1, 2, 3, 3]

Explanation RecentCounter recentCounter = new RecentCounter(); recentCounter.ping(1); // requests = [1], range is [-2999,1], return 1 recentCounter.ping(100); // requests = [1, 100], range is [-2900,100], return 2 recentCounter.ping(3001); // requests = [1, 100, 3001], range is [1,3001], return

3 recentCounter.ping(3002); // requests = [1, 100, 3001, 3002], range is [2,3002], return 3

Constraints:

1 <= t <= 109 Each test case will call ping with strictly increasing values of t. At most 104 calls will be made to ping.

==
Problem Number: 185 URL: https://leetcode.com/problems/range-sum-of-bst Title: 938. Range Sum of BST Problem Description: Given the root node of a binary search tree and two integers low and high, return the sum of values of all nodes with a value in the inclusive range [low, high]. Example 1:

Input: root = [10,5,15,3,7,null,18], low = 7, high = 15 Output: 32 Explanation: Nodes 7, 10, and 15 are in the range [7, 15]. 7 + 10 + 15 = 32.

Example 2:

Input: root = [10,5,15,3,7,13,18,1,null,6], low = 6, high = 10 Output: 23 Explanation: Nodes 6, 7, and 10 are in the range [6, 10]. 6 + 7 + 10 = 23.

Constraints:

The number of nodes in the tree is in the range [1, 2 * 104]. 1 <= Node.val <= 105 1 <= low <= high <= 105 All Node.val are unique.

==
Problem Number: 186 URL: https://leetcode.com/problems/valid-mountain-array Title: 941. Valid Mountain Array Problem Description: Given an array of integers arr, return true if and only if it is a valid mountain array. Recall that arr is a mountain array if and only if:

arr.length >= 3 There exists some i with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i] arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Example 1: Input: arr = [2,1] Output: false Example 2: Input: arr = [3,5,5] Output: false Example 3: Input: arr = [0,3,2,1] Output: true

Constraints:

1 <= arr.length <= 104 0 <= arr[i] <= 104

==
Problem Number: 187 URL: https://leetcode.com/problems/di-string-match Title: 942. DI String Match Problem Description: A permutation perm of n + 1 integers of all the integers in the range [0, n] can be represented as a string s of length n where:

s[i] == 'I' if perm[i] < perm[i + 1], and s[i] == 'D' if perm[i] > perm[i + 1].

Given a string s, reconstruct the permutation perm and return it. If there are multiple valid permutations perm, return any of them. Example 1: Input: s = "IDID" Output: [0,4,1,3,2] Example 2: Input: s = "III" Output: [0,1,2,3] Example 3: Input: s = "DDI" Output: [3,2,0,1]

Constraints:

1 <= s.length <= 105 s[i] is either 'I' or 'D'.

==
Problem Number: 188 URL: https://leetcode.com/problems/delete-columns-to-make-sorted Title: 944. Delete Columns to Make Sorted Problem Description: You are given an array of n strings strs, all of the same length. The strings can be arranged such that there is one on each line, making a grid.

For example, strs = ["abc", "bce", "cae"] can be arranged as follows:

abc bce cae

You want to delete the columns that are not sorted lexicographically. In the above example (0-indexed), columns 0 ('a', 'b', 'c') and 2 ('c', 'e', 'e') are sorted, while column 1 ('b', 'c', 'a') is not, so you would delete column 1. Return the number of columns that you will delete. Example 1: Input: strs = ["cba","daf","ghi"] Output: 1 Explanation: The grid looks as follows: cba daf ghi Columns 0 and 2 are sorted, but column 1 is not, so you only need to delete 1 column.

Example 2: Input: strs = ["a","b"] Output: 0 Explanation: The grid looks as follows: a b Column 0 is the only column and is sorted, so you will not delete any columns.

Example 3: Input: strs = ["zyx","wvu","tsr"] Output: 3 Explanation: The grid looks as follows: zyx wvu tsr All 3 columns are not sorted, so you will delete all 3.

Constraints:

n == strs.length 1 <= n <= 100 1 <= strs[i].length <= 1000 strs[i] consists of lowercase English letters.

==
Problem Number: 189 URL: https://leetcode.com/problems/verifying-an-alien-dictionary Title: 953. Verifying an Alien Dictionary Problem Description: In an alien language, surprisingly, they also use English lowercase letters, but possibly in a different order. The order of the alphabet is some permutation of lowercase letters. Given a sequence of words written in the alien language, and the order of the alphabet, return true if and only if the given words are sorted lexicographically in this alien language. Example 1: Input: words = ["hello","leetcode"], order = "hlabcdefgijkmnopqrstuvwxyz" Output: true Explanation: As 'h' comes before 'l' in this language, then the sequence is sorted.

Example 2: Input: words = ["word","world","row"], order = "worldabcefghijkmnpqstuvxyz" Output: false Explanation: As 'd' comes after 'l' in this language, then words[0] > words[1], hence the sequence is unsorted.

Example 3: Input: words = ["apple","app"], order = "abcdefghijklmnopqrstuvwxyz" Output: false Explanation: The first three characters "app" match, and the second string is shorter (in size.) According to lexicographical rules "apple" > "app", because 'l' > ' ', where ' ' is defined as the blank character which is less than any other character (More info).

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 20 order.length == 26 All characters in words[i] and order are English lowercase letters.

==

Problem Number: 190 URL: https://leetcode.com/problems/n-repeated-element-in-size-2n-array Title: 961. N-Repeated Element in Size 2N Array Problem Description: You are given an integer array nums with the following properties:

nums.length == 2 * n. nums contains n + 1 unique elements. Exactly one element of nums is repeated n times.

Return the element that is repeated n times. Example 1: Input: nums = [1,2,3,3] Output: 3 Example 2: Input: nums = [2,1,2,5,3,2] Output: 2 Example 3: Input: nums = [5,1,5,2,5,3,5,4] Output: 5

Constraints:

2 <= n <= 5000 nums.length == 2 * n 0 <= nums[i] <= 104 nums contains n + 1 unique elements and one of them is repeated exactly n times.

==

Problem Number: 191 URL: https://leetcode.com/problems/univalued-binary-tree Title: 965. Univalued Binary Tree Problem Description: A binary tree is uni-valued if every node in the tree has the same value. Given the root of a binary tree, return true if the given tree is uni-valued, or false otherwise. Example 1:

Input: root = [1,1,1,1,1,null,1] Output: true

Example 2:

Input: root = [2,2,2,5,2] Output: false

Constraints:

The number of nodes in the tree is in the range [1, 100]. 0 <= Node.val < 100

==

Problem Number: 192 URL: https://leetcode.com/problems/largest-perimeter-triangle Title: 976. Largest Perimeter Triangle Problem Description: Given an integer array nums, return the largest perimeter of a triangle with a non-zero area, formed from three of these lengths. If it is impossible to form any triangle of a non-zero area, return 0. Example 1: Input: nums = [2,1,2] Output: 5 Explanation: You can form a triangle with three side lengths: 1, 2, and 2.

Example 2: Input: nums = [1,2,1,10] Output: 0 Explanation: You cannot use the side lengths 1, 1, and 2 to form a triangle. You cannot use the side lengths 1, 1, and 10 to form a triangle. You cannot use the side lengths 1, 2, and 10 to form a triangle. As we cannot use any three side lengths to form a triangle of non-zero area, we return 0.

Constraints:

3 <= nums.length <= 104 1 <= nums[i] <= 106

==

Problem Number: 193 URL: https://leetcode.com/problems/squares-of-a-sorted-array Title: 977. Squares of a Sorted Array Problem Description: Given an integer array nums sorted in non-decreasing order, return an array of the squares of each number sorted in non-decreasing order. Example 1: Input: nums = [-4,-1,0,3,10] Output: [0,1,9,16,100] Explanation: After squaring, the array becomes [16,1,0,9,100]. After sorting, it becomes [0,1,9,16,100].

Example 2: Input: nums = [-7,-3,2,3,11] Output: [4,9,9,49,121]

Constraints:

1 <= nums.length <= 104 -104 <= nums[i] <= 104 nums is sorted in non-decreasing order.

Follow up: Squaring each element and sorting the new array is very trivial, could you find an O(n) solution using a different approach?
==

Problem Number: 194 URL: https://leetcode.com/problems/add-to-array-form-of-integer Title: 989. Add to Array-Form of Integer Problem Description: The array-form of an integer num is an array representing its digits in left to right order.

For example, for num = 1321, the array form is [1,3,2,1].

Given num, the array-form of an integer, and an integer k, return the array-form of the integer num + k. Example 1: Input: num = [1,2,0,0], k = 34 Output: [1,2,3,4] Explanation: 1200 + 34 = 1234

Example 2: Input: num = [2,7,4], k = 181 Output: [4,5,5] Explanation: 274 + 181 = 455

Example 3: Input: num = [2,1,5], k = 806 Output: [1,0,2,1] Explanation: 215 + 806 = 1021

Constraints:

1 <= num.length <= 104 0 <= num[i] <= 9 num does not contain any leading zeros except for the zero itself. 1 <= k <= 104

==
Problem Number: 195 URL: https://leetcode.com/problems/cousins-in-binary-tree Title: 993. Cousins in Binary Tree Problem Description: Given the root of a binary tree with unique values and the values of two different nodes of the tree x and y, return true if the nodes corresponding to the values x and y in the tree are cousins, or false otherwise. Two nodes of a binary tree are cousins if they have the same depth with different parents. Note that in a binary tree, the root node is at the depth 0, and children of each depth k node are at the depth k + 1.   Example 1:

Input: root = [1,2,3,4], x = 4, y = 3 Output: false

Example 2:

Input: root = [1,2,3,null,4,null,5], x = 5, y = 4 Output: true

Example 3:

Input: root = [1,2,3,null,4], x = 2, y = 3 Output: false

Constraints:

The number of nodes in the tree is in the range [2, 100]. 1 <= Node.val <= 100 Each node has a unique value. x != y x and y are exist in the tree.

==
Problem Number: 196 URL: https://leetcode.com/problems/find-the-town-judge Title: 997. Find the Town Judge Problem Description: In a town, there are n people labeled from 1 to n. There is a rumor that one of these people is secretly the town judge. If the town judge exists, then:

The town judge trusts nobody. Everybody (except for the town judge) trusts the town judge. There is exactly one person that satisfies properties 1 and 2.

You are given an array trust where trust[i] = [ai, bi] representing that the person labeled ai trusts the person labeled bi. If a trust relationship does not exist in trust array, then such a trust relationship does not exist. Return the label of the town judge if the town judge exists and can be identified, or return -1 otherwise. Example 1: Input: n = 2, trust = [[1,2]] Output: 2

Example 2: Input: n = 3, trust = [[1,3],[2,3]] Output: 3

Example 3: Input: n = 3, trust = [[1,3],[2,3],[3,1]] Output: -1

Constraints:

1 <= n <= 1000 0 <= trust.length <= 104 trust[i].length == 2 All the pairs of trust are unique. ai != bi 1 <= ai, bi <= n

==
Problem Number: 197 URL: https://leetcode.com/problems/available-captures-

for-rook Title: 999. Available Captures for Rook Problem Description: On an 8 x 8 chessboard, there is exactly one white rook ’R’ and some number of white bishops ’B’, black pawns ’p’, and empty squares ’.’. When the rook moves, it chooses one of four cardinal directions (north, east, south, or west), then moves in that direction until it chooses to stop, reaches the edge of the board, captures a black pawn, or is blocked by a white bishop. A rook is considered attacking a pawn if the rook can capture the pawn on the rook’s turn. The number of available captures for the white rook is the number of pawns that the rook is attacking. Return the number of available captures for the white rook. Example 1:

Input: board = [[”.”,”.”,”.”,”.”,”.”,”.”,”.”,”.”],[”.”,”.”,”.”,”p”,”.”,”.”,”.”,”.”],[”.”,”.”,”.”,”R”,”.”,”.”,”.”,”p”],[”.”,”.”,”.”,”.”,”.”,”.”,”
Output: 3 Explanation: In this example, the rook is attacking all the pawns.

Example 2:

Input: board = [[”.”,”.”,”.”,”.”,”.”,”.”,”.”,”.”],[”.”,”p”,”p”,”p”,”p”,”p”,”.”,”.”],[”.”,”p”,”p”,”B”,”p”,”p”,”.”,”.”],[”.”,”p”,”B
Output: 0 Explanation: The bishops are blocking the rook from attacking any of the pawns.

Example 3:

Input: board = [[”.”,”.”,”.”,”.”,”.”,”.”,”.”,”.”],[”.”,”.”,”.”,”p”,”.”,”.”,”.”,”.”],[”.”,”.”,”.”,”p”,”.”,”.”,”.”,”.”],[”p”,”p”,”.”,”R”,”.”,”
Output: 3 Explanation: The rook is attacking the pawns at positions b5, d6, and f5.

Constraints:

board.length == 8 board[i].length == 8 board[i][j] is either ’R’, ’.’, ’B’, or ’p’ There is exactly one cell with board[i][j] == ’R’

===
Problem Number: 198 URL: https://leetcode.com/problems/find-common-characters Title: 1002. Find Common Characters Problem Description: Given a string array words, return an array of all characters that show up in all strings within the words (including duplicates). You may return the answer in any order. Example 1: Input: words = [”bella”,”label”,”roller”] Output: [”e”,”l”,”l”] Example 2: Input: words = [”cool”,”lock”,”cook”] Output: [”c”,”o”]

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 100 words[i] consists of lowercase English letters.

===
Problem Number: 199 URL: https://leetcode.com/problems/maximize-sum-of-array-after-k-negations Title: 1005. Maximize Sum Of Array After K Negations Problem Description: Given an integer array nums and an integer k, modify the array in the following way:

choose an index i and replace nums[i] with -nums[i].

You should apply this process exactly k times. You may choose the same index i multiple times. Return the largest possible sum of the array after modifying it in this way. Example 1: Input: nums = [4,2,3], k = 1 Output: 5 Explanation: Choose index 1 and nums becomes [4,-2,3].

Example 2: Input: nums = [3,-1,0,2], k = 3 Output: 6 Explanation: Choose indices (1, 2, 2) and nums becomes [3,1,0,2].

Example 3: Input: nums = [2,-3,-1,5,-4], k = 2 Output: 13 Explanation: Choose indices (1, 4) and nums becomes [2,3,-1,5,4].

Constraints:

1 <= nums.length <= 104 -100 <= nums[i] <= 100 1 <= k <= 104

==
Problem Number: 200 URL: https://leetcode.com/problems/complement-of-base-10-integer Title: 1009. Complement of Base 10 Integer Problem Description: The complement of an integer is the integer you get when you flip all the 0's to 1's and all the 1's to 0's in its binary representation.

For example, The integer 5 is "101" in binary and its complement is "010" which is the integer 2.

Given an integer n, return its complement. Example 1: Input: n = 5 Output: 2 Explanation: 5 is "101" in binary, with complement "010" in binary, which is 2 in base-10.

Example 2: Input: n = 7 Output: 0 Explanation: 7 is "111" in binary, with complement "000" in binary, which is 0 in base-10.

Example 3: Input: n = 10 Output: 5 Explanation: 10 is "1010" in binary, with complement "0101" in binary, which is 5 in base-10.

Constraints:

0 <= n < 109

Note: This question is the same as 476: https://leetcode.com/problems/number-complement/

==
Problem Number: 201 URL: https://leetcode.com/problems/partition-array-into-three-parts-with-equal-sum Title: 1013. Partition Array Into Three Parts With Equal Sum Problem Description: Given an array of integers arr, return true if we can partition the array into three non-empty parts with equal sums. Formally, we can partition the array if we can find indexes i + 1 < j with (arr[0] + arr[1] + ... + arr[i] == arr[i + 1] + arr[i + 2] + ... + arr[j - 1] == arr[j] + arr[j + 1] + ... + arr[arr.length - 1]) Example 1: Input: arr = [0,2,1,-6,6,-7,9,1,2,0,1] Output: true Explanation: 0 + 2 + 1 = -6 + 6 - 7 + 9 + 1 = 2 + 0 + 1

Example 2: Input: arr = [0,2,1,-6,6,7,9,-1,2,0,1] Output: false

Example 3: Input: arr = [3,3,6,5,-2,2,5,1,-9,4] Output: true Explanation: 3 + 3 = 6 = 5 - 2 + 2 + 5 + 1 - 9 + 4

Constraints:

3 <= arr.length <= 5 * 104 -104 <= arr[i] <= 104

==
Problem Number: 202 URL: https://leetcode.com/problems/binary-prefix-divisible-by-5 Title: 1018. Binary Prefix Divisible By 5 Problem Description: You are given a binary array nums (0-indexed). We define xi as the number whose binary representation is the subarray nums[0..i] (from most-significant-bit to least-significant-bit).

For example, if nums = [1,0,1], then x0 = 1, x1 = 2, and x2 = 5.

Return an array of booleans answer where answer[i] is true if xi is divisible by 5. Example 1: Input: nums = [0,1,1] Output: [true,false,false] Explanation: The input numbers in binary are 0, 01, 011; which are 0, 1, and 3 in base-10. Only the first number is divisible by 5, so answer[0] is true.

Example 2: Input: nums = [1,1,1] Output: [false,false,false]

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1.

==
Problem Number: 203 URL: https://leetcode.com/problems/remove-outermost-parentheses Title: 1021. Remove Outermost Parentheses Problem Description: A valid parentheses string is either empty "", "(" + A + ")", or A + B, where A and B are valid parentheses strings, and + represents string concatenation.

For example, "", "()", "(())()", and "(()(()))" are all valid parentheses strings.

A valid parentheses string s is primitive if it is nonempty, and there does not exist a way to split it into s = A + B, with A and B nonempty valid parentheses strings. Given a valid parentheses string s, consider its primitive decomposition: s = P1 + P2 + ... + Pk, where Pi are primitive valid parentheses strings. Return s after removing the outermost parentheses of every primitive string in the primitive decomposition of s. Example 1: Input: s = "(()())(())" Output: "()()()" Explanation: The input string is "(()())(())", with primitive decomposition "(()())" + "(())". After removing outer parentheses of each part, this is "()()" + "()" = "()()()".

Example 2: Input: s = "(()())(())(()(()))" Output: "()()()()(())" Explanation: The input string is "(()())(())(()(()))", with primitive decomposition "(()())" + "(())" + "(()(()))". After removing outer parentheses of each part, this is "()()" + "()" + "()(())" = "()()()()(())".

Example 3: Input: s = "()()" Output: "" Explanation: The input string is "()()", with primitive decomposition "()" + "()". After removing outer parentheses of each part, this is "" + "" = "".

Constraints:

1 <= s.length <= 105 s[i] is either '(' or ')'. s is a valid parentheses string.

==

Problem Number: 204 URL: https://leetcode.com/problems/sum-of-root-to-leaf-binary-numbers Title: 1022. Sum of Root To Leaf Binary Numbers Problem Description: You are given the root of a binary tree where each node has a value 0 or 1. Each root-to-leaf path represents a binary number starting with the most significant bit.

For example, if the path is 0 -> 1 -> 1 -> 0 -> 1, then this could represent 01101 in binary, which is 13.

For all leaves in the tree, consider the numbers represented by the path from the root to that leaf. Return the sum of these numbers. The test cases are generated so that the answer fits in a 32-bits integer. Example 1:

Input: root = [1,0,1,0,1,0,1] Output: 22 Explanation: (100) + (101) + (110) + (111) = 4 + 5 + 6 + 7 = 22

Example 2: Input: root = [0] Output: 0

Constraints:

The number of nodes in the tree is in the range [1, 1000]. Node.val is 0 or 1.

==

Problem Number: 205 URL: https://leetcode.com/problems/divisor-game Title: 1025. Divisor Game Problem Description: Alice and Bob take turns playing a game, with Alice starting first. Initially, there is a number n on the chalkboard. On each player's turn, that player makes a move consisting of:

Choosing any x with 0 < x < n and n % x == 0. Replacing the number n on the chalkboard with n - x.

Also, if a player cannot make a move, they lose the game. Return true if and only if Alice wins the game, assuming both players play optimally. Example 1: Input: n = 2 Output: true Explanation: Alice chooses 1, and Bob has no more moves.

Example 2: Input: n = 3 Output: false Explanation: Alice chooses 1, Bob chooses 1, and Alice has no more moves.

Constraints:

1 <= n <= 1000

==

Problem Number: 206 URL: https://leetcode.com/problems/matrix-cells-in-distance-order Title: 1030. Matrix Cells in Distance Order Problem Description: You are given four integers row, cols, rCenter, and cCenter. There is a rows x cols matrix and you are on the cell with the coordinates (rCenter, cCenter). Return the coordinates of all cells in the matrix, sorted by their distance from (rCenter, cCenter) from the smallest distance to the largest distance. You may return the answer in any order that satisfies this condition. The distance between two cells (r1, c1) and (r2, c2) is |r1 - r2| + |c1 - c2|. Example 1: Input: rows = 1, cols = 2, rCenter = 0, cCenter = 0 Output: [[0,0],[0,1]] Explanation: The distances from (0, 0) to other cells are: [0,1]

Example 2: Input: rows = 2, cols = 2, rCenter = 0, cCenter = 1 Output: [[0,1],[0,0],[1,1],[1,0]] Explanation: The distances from (0, 1) to other cells are: [0,1,1,2] The answer [[0,1],[1,1],[0,0],[1,0]] would also be accepted as correct.

Example 3: Input: rows = 2, cols = 3, rCenter = 1, cCenter = 2 Output: [[1,2],[0,2],[1,1],[0,1],[1,0],[0,0]] Explanation: The distances from (1, 2) to other cells are: [0,1,1,2,2,3] There are other answers that would also be accepted as correct, such as [[1,2],[1,1],[0,2],[1,0],[0,1],[0,0]].

Constraints:

1 <= rows, cols <= 100 0 <= rCenter < rows 0 <= cCenter < cols
==
Problem Number: 207 URL: https://leetcode.com/problems/valid-boomerang Title: 1037. Valid Boomerang Problem Description: Given an array points where points[i] = [xi, yi] represents a point on the X-Y plane, return true if these points are a boomerang. A boomerang is a set of three points that are all distinct and not in a straight line. Example 1: Input: points = [[1,1],[2,3],[3,2]] Output: true Example 2: Input: points = [[1,1],[2,2],[3,3]] Output: false

Constraints:

points.length == 3 points[i].length == 2 0 <= xi, yi <= 100
==
Problem Number: 208 URL: https://leetcode.com/problems/last-stone-weight Title: 1046. Last Stone Weight Problem Description: You are given an array of integers stones where stones[i] is the weight of the ith stone. We are playing a game with the stones. On each turn, we choose the heaviest two stones and smash them together. Suppose the heaviest two stones have weights x and y with x <= y. The result of this smash is:

If x == y, both stones are destroyed, and If x != y, the stone of weight x is destroyed, and the stone of weight y has new weight y - x.

At the end of the game, there is at most one stone left. Return the weight of the last remaining stone. If there are no stones left, return 0. Example 1: Input: stones = [2,7,4,1,8,1] Output: 1 Explanation: We combine 7 and 8 to get 1 so the array converts to [2,4,1,1,1] then, we combine 2 and 4 to get 2 so the array converts to [2,1,1,1] then, we combine 2 and 1 to get 1 so the array converts to [1,1,1] then, we combine 1 and 1 to get 0 so the array converts to [1] then that's the value of the last stone.

Example 2: Input: stones = [1] Output: 1

Constraints:

1 <= stones.length <= 30 1 <= stones[i] <= 1000

==
Problem Number: 209 URL: https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string Title: 1047. Remove All Adjacent Duplicates In String Problem Description: You are given a string s consisting of lowercase English letters. A duplicate removal consists of choosing two adjacent and equal letters and removing them. We repeatedly make duplicate removals on s until we no longer can. Return the final string after all such duplicate removals have been made. It can be proven that the answer is unique. Example 1: Input: s = "abbaca" Output: "ca" Explanation: For example, in "abbaca" we could remove "bb" since the letters are adjacent and equal, and this is the only possible move. The result of this move is that the string is "aaca", of which only "aa" is possible, so the final string is "ca".

Example 2: Input: s = "azxxzy" Output: "ay"

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters.

==
Problem Number: 210 URL: https://leetcode.com/problems/height-checker Title: 1051. Height Checker Problem Description: A school is trying to take an annual photo of all the students. The students are asked to stand in a single file line in non-decreasing order by height. Let this ordering be represented by the integer array expected where expected[i] is the expected height of the ith student in line. You are given an integer array heights representing the current order that the students are standing in. Each heights[i] is the height of the ith student in line (0-indexed). Return the number of indices where heights[i] != expected[i]. Example 1: Input: heights = [1,1,4,2,1,3] Output: 3 Explanation: heights: [1,1,4,2,1,3] expected: [1,1,1,2,3,4] Indices 2, 4, and 5 do not match.

Example 2: Input: heights = [5,1,2,3,4] Output: 5 Explanation: heights: [5,1,2,3,4] expected: [1,2,3,4,5] All indices do not match.

Example 3: Input: heights = [1,2,3,4,5] Output: 0 Explanation: heights: [1,2,3,4,5] expected: [1,2,3,4,5] All indices match.

Constraints:

1 <= heights.length <= 100 1 <= heights[i] <= 100

==
Problem Number: 211 URL: https://leetcode.com/problems/greatest-common-divisor-of-strings Title: 1071. Greatest Common Divisor of Strings Problem Description: For two strings s and t, we say "t divides s" if and only if s = t + ... + t (i.e., t is concatenated with itself one or more times). Given two strings str1 and str2, return the largest string x such that x divides both str1 and str2. Example 1: Input: str1 = "ABCABC", str2 = "ABC" Output: "ABC"

Example 2: Input: str1 = "ABABAB", str2 = "ABAB" Output: "AB"

Example 3: Input: str1 = "LEET", str2 = "CODE" Output: ""

Constraints:

1 <= str1.length, str2.length <= 1000 str1 and str2 consist of English uppercase letters.

==
Problem Number: 212 URL: https://leetcode.com/problems/occurrences-after-bigram Title: 1078. Occurrences After Bigram Problem Description: Given two strings first and second, consider occurrences in some text of the form "first second third", where second comes immediately after first, and third comes immediately after second. Return an array of all the words third for each occurrence of "first second third". Example 1: Input: text = "alice is a good girl she is a good student", first = "a", second = "good" Output: ["girl","student"] Example 2: Input: text = "we will we will rock you", first = "we", second = "will" Output: ["we","rock"]

Constraints:

1 <= text.length <= 1000 text consists of lowercase English letters and spaces. All the words in text a separated by a single space. 1 <= first.length, second.length <= 10 first and second consist of lowercase English letters.

==
Problem Number: 213 URL: https://leetcode.com/problems/duplicate-zeros Title: 1089. Duplicate Zeros Problem Description: Given a fixed-length integer array arr, duplicate each occurrence of zero, shifting the remaining elements to the right. Note that elements beyond the length of the original array are not written. Do the above modifications to the input array in place and do not return anything. Example 1: Input: arr = [1,0,2,3,0,4,5,0] Output: [1,0,0,2,3,0,0,4] Explanation: After calling your function, the input array is modified to: [1,0,0,2,3,0,0,4]

Example 2: Input: arr = [1,2,3] Output: [1,2,3] Explanation: After calling your function, the input array is modified to: [1,2,3]

Constraints:

1 <= arr.length <= 104 0 <= arr[i] <= 9

==
Problem Number: 214 URL: https://leetcode.com/problems/distribute-candies-to-people Title: 1103. Distribute Candies to People Problem Description: We distribute some number of candies, to a row of n = num_people people in the following way: We then give 1 candy to the first person, 2 candies to the second person, and so on until we give n candies to the last person. Then, we go back to the start of the row, giving n + 1 candies to the first person, n + 2 candies to the second person, and so on until we give 2 * n candies to the last person. This process repeats (with us giving one more candy each time, and moving to the start of the row after we reach the end) until we run out of candies. The last person will receive all of our remaining candies (not necessarily one more than the previous gift). Return an array (of length num_people and sum candies) that represents the final distribution of candies. Example 1: Input: candies = 7, num_people = 4 Output: [1,2,3,1] Explanation: On the first turn, ans[0] += 1, and the array is [1,0,0,0]. On the second turn, ans[1] += 2, and the array is [1,2,0,0]. On the third turn, ans[2] += 3, and the array is [1,2,3,0]. On the fourth turn, ans[3] += 1 (because there is only one candy left), and the final array is [1,2,3,1].

Example 2: Input: candies = 10, num_people = 3 Output: [5,2,3] Explanation: On the first turn, ans[0] += 1, and the array is [1,0,0]. On the second turn, ans[1] += 2, and the array is [1,2,0]. On the third turn, ans[2] += 3, and the array is [1,2,3]. On the fourth turn, ans[0] += 4, and the final array is [5,2,3].

Constraints:

1 <= candies <= 10^9 1 <= num_people <= 1000

==
Problem Number: 215 URL: https://leetcode.com/problems/defanging-an-ip-address Title: 1108. Defanging an IP Address Problem Description: Given a valid (IPv4) IP address, return a defanged version of that IP address. A defanged IP address replaces every period "." with "[.]". Example 1: Input: address = "1.1.1.1" Output: "1[.]1[.]1[.]1" Example 2: Input: address = "255.100.50.0" Output: "255[.]100[.]50[.]0"

Constraints:

The given address is a valid IPv4 address.

==
Problem Number: 216 URL: https://leetcode.com/problems/relative-sort-array Title: 1122. Relative Sort Array Problem Description: Given two arrays arr1 and arr2, the elements of arr2 are distinct, and all elements in arr2 are also in arr1. Sort the elements of arr1 such that the relative ordering of items in arr1 are the same as in arr2. Elements that do not appear in arr2 should be placed at the end of arr1 in ascending order. Example 1: Input: arr1 = [2,3,1,3,2,4,6,7,9,2,19], arr2 = [2,1,4,3,9,6] Output: [2,2,2,1,4,3,3,9,6,7,19]

Example 2: Input: arr1 = [28,6,22,8,44,17], arr2 = [22,28,8,6] Output: [22,28,8,6,17,44]

Constraints:

1 <= arr1.length, arr2.length <= 1000 0 <= arr1[i], arr2[i] <= 1000 All the elements of arr2 are distinct. Each arr2[i] is in arr1.

==
Problem Number: 217 URL: https://leetcode.com/problems/number-of-equivalent-domino-pairs Title: 1128. Number of Equivalent Domino Pairs Problem Description: Given a list of dominoes, dominoes[i] = [a, b] is equivalent to dominoes[j] = [c, d] if and only if either (a == c and b == d), or (a == d and b == c) - that is, one domino can be rotated to be equal to another domino. Return the number of pairs (i, j) for which 0 <= i < j < dominoes.length, and dominoes[i] is equivalent to dominoes[j]. Example 1: Input: dominoes = [[1,2],[2,1],[3,4],[5,6]] Output: 1

Example 2: Input: dominoes = [[1,2],[1,2],[1,1],[1,2],[2,2]] Output: 3

Constraints:

1 <= dominoes.length <= 4 * 104 dominoes[i].length == 2 1 <= dominoes[i][j] <= 9

==
Problem Number: 218 URL: https://leetcode.com/problems/n-th-tribonacci-number Title: 1137. N-th Tribonacci Number Problem Description: The Tribonacci sequence Tn is defined as follows: T0 = 0, T1 = 1, T2 = 1, and Tn+3 = Tn + Tn+1 + Tn+2 for n >= 0. Given n, return the value of Tn. Example 1: Input: n = 4 Output: 4 Explanation: T_3 = 0 + 1 + 1 = 2 T_4 = 1 + 1 + 2 = 4

Example 2: Input: n = 25 Output: 1389537

Constraints:

0 <= n <= 37 The answer is guaranteed to fit within a 32-bit integer, ie. answer <= 2^31 - 1.

==
Problem Number: 219 URL: https://leetcode.com/problems/day-of-the-year Title: 1154. Day of the Year Problem Description: Given a string date representing a Gregorian calendar date formatted as YYYY-MM-DD, return the day number of the year. Example 1: Input: date = "2019-01-09" Output: 9 Explanation: Given date is the 9th day of the year in 2019.

Example 2: Input: date = "2019-02-10" Output: 41

Constraints:

date.length == 10 date[4] == date[7] == '-', and all other date[i]'s are digits date represents a calendar date between Jan 1st, 1900 and Dec 31th, 2019.

===
Problem Number: 220 URL: https://leetcode.com/problems/find-words-that-can-be-formed-by-characters Title: 1160. Find Words That Can Be Formed by Characters Problem Description: You are given an array of strings words and a string chars. A string is good if it can be formed by characters from chars (each character can only be used once). Return the sum of lengths of all good strings in words. Example 1: Input: words = [”cat”,”bt”,”hat”,”tree”], chars = ”atach” Output: 6 Explanation: The strings that can be formed are ”cat” and ”hat” so the answer is 3 + 3 = 6.

Example 2: Input: words = [”hello”,”world”,”leetcode”], chars = ”welldonehoneyr” Output: 10 Explanation: The strings that can be formed are ”hello” and ”world” so the answer is 5 + 5 = 10.

Constraints:

1 <= words.length <= 1000 1 <= words[i].length, chars.length <= 100 words[i] and chars consist of lowercase English letters.

===
Problem Number: 221 URL: https://leetcode.com/problems/prime-arrangements Title: 1175. Prime Arrangements Problem Description: Return the number of permutations of 1 to n so that prime numbers are at prime indices (1-indexed.) (Recall that an integer is prime if and only if it is greater than 1, and cannot be written as a product of two positive integers both smaller than it.) Since the answer may be large, return the answer modulo 10^9 + 7. Example 1: Input: n = 5 Output: 12 Explanation: For example [1,2,5,4,3] is a valid permutation, but [5,2,3,4,1] is not because the prime number 5 is at index 1.

Example 2: Input: n = 100 Output: 682289015

Constraints:

1 <= n <= 100

===
Problem Number: 222 URL: https://leetcode.com/problems/distance-between-bus-stops Title: 1184. Distance Between Bus Stops Problem Description: A bus has n stops numbered from 0 to n - 1 that form a circle. We know the distance between all pairs of neighboring stops where distance[i] is the distance between the stops number i and (i + 1) % n. The bus goes along both directions i.e. clockwise and counterclockwise. Return the shortest distance between the given start and destination stops. Example 1:

Input: distance = [1,2,3,4], start = 0, destination = 1 Output: 1 Explanation: Distance between 0 and 1 is 1 or 9, minimum is 1. Example 2:

Input: distance = [1,2,3,4], start = 0, destination = 2 Output: 3 Explanation: Distance between 0 and 2 is 3 or 7, minimum is 3.

Example 3:

Input: distance = [1,2,3,4], start = 0, destination = 3 Output: 4 Explanation: Distance between 0 and 3 is 6 or 4, minimum is 4.

Constraints:

1 <= n <= 10^4 distance.length == n 0 <= start, destination < n 0 <= distance[i] <= 10^4

==
Problem Number: 223 URL: https://leetcode.com/problems/day-of-the-week Title: 1185. Day of the Week Problem Description: Given a date, return the corresponding day of the week for that date. The input is given as three integers representing the day, month and year respectively. Return the answer as one of the following values {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}. Example 1: Input: day = 31, month = 8, year = 2019 Output: "Saturday"

Example 2: Input: day = 18, month = 7, year = 1999 Output: "Sunday"

Example 3: Input: day = 15, month = 8, year = 1993 Output: "Sunday"

Constraints:

The given dates are valid dates between the years 1971 and 2100.

==
Problem Number: 224 URL: https://leetcode.com/problems/maximum-number-of-balloons Title: 1189. Maximum Number of Balloons Problem Description: Given a string text, you want to use the characters of text to form as many instances of the word "balloon" as possible. You can use each character in text at most once. Return the maximum number of instances that can be formed. Example 1:

Input: text = "nlaebolko" Output: 1

Example 2:

Input: text = "loonbalxballpoon" Output: 2

Example 3: Input: text = "leetcode" Output: 0

Constraints:

1 <= text.length <= 104 text consists of lower case English letters only.

==
Problem Number: 225 URL: https://leetcode.com/problems/minimum-absolute-difference Title: 1200. Minimum Absolute Difference Problem Description: Given an array of distinct integers arr, find all pairs of elements with the minimum absolute difference of any two elements. Return a list of pairs in ascending order(with respect to pairs), each pair [a, b] follows

a, b are from arr a < b b - a equals to the minimum absolute difference of any two elements in arr

Example 1: Input: arr = [4,2,1,3] Output: [[1,2],[2,3],[3,4]] Explanation: The minimum absolute difference is 1. List all pairs with difference equal to 1 in ascending order. Example 2: Input: arr = [1,3,6,10,15] Output: [[1,3]]

Example 3: Input: arr = [3,8,-10,23,19,-4,-14,27] Output: [[-14,-10],[19,23],[23,27]]

Constraints:

2 <= arr.length <= 105 -106 <= arr[i] <= 106

==

Problem Number: 226 URL: https://leetcode.com/problems/unique-number-of-occurrences Title: 1207. Unique Number of Occurrences Problem Description: Given an array of integers arr, return true if the number of occurrences of each value in the array is unique or false otherwise. Example 1: Input: arr = [1,2,2,1,1,3] Output: true Explanation: The value 1 has 3 occurrences, 2 has 2 and 3 has 1. No two values have the same number of occurrences. Example 2: Input: arr = [1,2] Output: false

Example 3: Input: arr = [-3,0,1,-3,1,1,1,-3,10,0] Output: true

Constraints:

1 <= arr.length <= 1000 -1000 <= arr[i] <= 1000

==

Problem Number: 227 URL: https://leetcode.com/problems/minimum-cost-to-move-chips-to-the-same-position Title: 1217. Minimum Cost to Move Chips to The Same Position Problem Description: We have n chips, where the position of the ith chip is position[i]. We need to move all the chips to the same position. In one step, we can change the position of the ith chip from position[i] to:

position[i] + 2 or position[i] - 2 with cost = 0. position[i] + 1 or position[i] - 1 with cost = 1.

Return the minimum cost needed to move all the chips to the same position. Example 1:

Input: position = [1,2,3] Output: 1 Explanation: First step: Move the chip at position 3 to position 1 with cost = 0. Second step: Move the chip at position 2 to position 1 with cost = 1. Total cost is 1.

Example 2:

Input: position = [2,2,2,3,3] Output: 2 Explanation: We can move the two chips at position 3 to position 2. Each move has cost = 1. The total cost = 2.

Example 3: Input: position = [1,1000000000] Output: 1

Constraints:

1 <= position.length <= 100 1 <= position[i] <= 10^9

==
Problem Number: 228 URL: https://leetcode.com/problems/split-a-string-in-balanced-strings Title: 1221. Split a String in Balanced Strings Problem Description: Balanced strings are those that have an equal quantity of 'L' and 'R' characters. Given a balanced string s, split it into some number of substrings such that:

Each substring is balanced.

Return the maximum number of balanced strings you can obtain. Example 1: Input: s = "RLRRLLRLRL" Output: 4 Explanation: s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2: Input: s = "RLRRRLLRLL" Output: 2 Explanation: s can be split into "RL", "RRRLLRLL", each substring contains same number of 'L' and 'R'. Note that s cannot be split into "RL", "RR", "RL", "LR", "LL", because the 2nd and 5th substrings are not balanced. Example 3: Input: s = "LLLLRRRR" Output: 1 Explanation: s can be split into "LLLLRRRR".

Constraints:

2 <= s.length <= 1000 s[i] is either 'L' or 'R'. s is a balanced string.

==
Problem Number: 229 URL: https://leetcode.com/problems/check-if-it-is-a-straight-line Title: 1232. Check If It Is a Straight Line Problem Description: You are given an array coordinates, coordinates[i] = [x, y], where [x, y] represents the coordinate of a point. Check if these points make a straight line in the XY plane. Example 1:

Input: coordinates = [[1,2],[2,3],[3,4],[4,5],[5,6],[6,7]] Output: true

Example 2:

Input: coordinates = [[1,1],[2,2],[3,4],[4,5],[5,6],[7,7]] Output: false

Constraints:

2 <= coordinates.length <= 1000 coordinates[i].length == 2 -10^4 <= coordinates[i][0], coordinates[i][1] <= 10^4 coordinates contains no duplicate point.

==
Problem Number: 230 URL: https://leetcode.com/problems/cells-with-odd-values-in-a-matrix Title: 1252. Cells with Odd Values in a Matrix Problem Description: There is an m x n matrix that is initialized to all 0's. There is also a 2D array indices where each indices[i] = [ri, ci] represents a 0-indexed location to perform some increment operations on the matrix. For each location indices[i], do both of the following:

Increment all the cells on row ri. Increment all the cells on column ci.

Given m, n, and indices, return the number of odd-valued cells in the matrix after applying the increment to all locations in indices. Example 1:

Input: m = 2, n = 3, indices = [[0,1],[1,1]] Output: 6 Explanation: Initial matrix = [[0,0,0],[0,0,0]]. After applying first increment it becomes [[1,2,1],[0,1,0]]. The final matrix is [[1,3,1],[1,3,1]], which contains 6 odd numbers.

Example 2:

Input: m = 2, n = 2, indices = [[1,1],[0,0]] Output: 0 Explanation: Final matrix = [[2,2],[2,2]]. There are no odd numbers in the final matrix.

Constraints:

1 <= m, n <= 50 1 <= indices.length <= 100 0 <= ri < m 0 <= ci < n

Follow up: Could you solve this in O(n + m + indices.length) time with only O(n + m) extra space?

==
Problem Number: 231 URL: https://leetcode.com/problems/shift-2d-grid Title: 1260. Shift 2D Grid Problem Description: Given a 2D grid of size m x n and an integer k. You need to shift the grid k times. In one shift operation:

Element at grid[i][j] moves to grid[i][j + 1]. Element at grid[i][n - 1] moves to grid[i + 1][0]. Element at grid[m - 1][n - 1] moves to grid[0][0].

Return the 2D grid after applying shift operation k times. Example 1:

Input: grid = [[1,2,3],[4,5,6],[7,8,9]], k = 1 Output: [[9,1,2],[3,4,5],[6,7,8]]

Example 2:

Input: grid = [[3,8,1,9],[19,7,2,5],[4,6,11,10],[12,0,21,13]], k = 4 Output: [[12,0,21,13],[3,8,1,9],[19,7,2,5],[4,6,11,10]]

Example 3: Input: grid = [[1,2,3],[4,5,6],[7,8,9]], k = 9 Output: [[1,2,3],[4,5,6],[7,8,9]]

Constraints:

m == grid.length n == grid[i].length 1 <= m <= 50 1 <= n <= 50 -1000 <= grid[i][j] <= 1000 0 <= k <= 100

==
Problem Number: 232 URL: https://leetcode.com/problems/minimum-time-visiting-all-points Title: 1266. Minimum Time Visiting All Points Problem Description: On a 2D plane, there are n points with integer coordinates points[i] = [xi, yi]. Return the minimum time in seconds to visit all the points in the order given by points. You can move according to these rules:

In 1 second, you can either:

move vertically by one unit, move horizontally by one unit, or move diagonally sqrt(2) units (in other words, move one unit vertically then one unit horizontally in 1 second).

You have to visit the points in the same order as they appear in the array. You are allowed to pass through points that appear later in the order, but these do not count as visits.

Example 1:

Input: points = [[1,1],[3,4],[-1,0]] Output: 7 Explanation: One optimal path is [1,1] -> [2,2] -> [3,3] -> [3,4] -> [2,3] -> [1,2] -> [0,1] -> [-1,0] Time from [1,1] to [3,4] = 3 seconds Time from [3,4] to [-1,0] = 4 seconds Total time = 7 seconds Example 2: Input: points = [[3,2],[-2,2]] Output: 5

Constraints:

points.length == n 1 <= n <= 100 points[i].length == 2 -1000 <= points[i][0], points[i][1] <= 1000

==
Problem Number: 233 URL: https://leetcode.com/problems/find-winner-on-a-tic-tac-toe-game Title: 1275. Find Winner on a Tic Tac Toe Game Problem Description: Tic-tac-toe is played by two players A and B on a 3 x 3 grid. The rules of Tic-Tac-Toe are:

Players take turns placing characters into empty squares ' '. The first player A always places 'X' characters, while the second player B always places 'O' characters. 'X' and 'O' characters are always placed into empty squares, never on filled ones. The game ends when there are three of the same (non-empty) character filling any row, column, or diagonal. The game also ends if all squares are non-empty. No more moves can be played if the game is over.

Given a 2D integer array moves where moves[i] = [rowi, coli] indicates that the ith move will be played on grid[rowi][coli]. return the winner of the game if it exists (A or B). In case the game ends in a draw return "Draw". If there are still movements to play return "Pending". You can assume that moves is valid (i.e., it follows the rules of Tic-Tac-Toe), the grid is initially empty, and A will play first.   Example 1:

Input: moves = [[0,0],[2,0],[1,1],[2,1],[2,2]] Output: "A" Explanation: A wins, they always play first.

Example 2:

Input: moves = [[0,0],[1,1],[0,1],[0,2],[1,0],[2,0]] Output: "B" Explanation: B wins.

Example 3:

Input: moves = [[0,0],[1,1],[2,0],[1,0],[1,2],[2,1],[0,1],[0,2],[2,2]] Output: "Draw" Explanation: The game ends in a draw since there are no moves to make.

Constraints:

1 <= moves.length <= 9 moves[i].length == 2 0 <= rowi, coli <= 2 There are no repeated elements on moves. moves follow the rules of tic tac toe.

==
Problem Number: 234 URL: https://leetcode.com/problems/subtract-the-product-and-sum-of-digits-of-an-integer Title: 1281. Subtract the Product and Sum of Digits of an Integer Problem Description: Given an integer number n, return the difference between the product of its digits and the sum of its digits. Example 1: Input: n = 234 Output: 15 Explanation: Product of digits = 2 * 3 * 4 = 24 Sum of digits = 2 + 3 + 4 = 9 Result = 24 - 9 = 15

Example 2: Input: n = 4421 Output: 21 Explanation: Product of digits = 4 * 4 * 2 * 1 = 32 Sum of digits = 4 + 4 + 2 + 1 = 11 Result = 32 - 11 = 21

Constraints:

1 <= n <= 10^5

==
Problem Number: 235 URL: https://leetcode.com/problems/element-appearing-more-than-25-in-sorted-array Title: 1287. Element Appearing More Than 25% In Sorted Array Problem Description: Given an integer array sorted in non-decreasing order, there is exactly one integer in the array that occurs more than 25% of the time, return that integer. Example 1: Input: arr = [1,2,2,6,6,6,6,7,10] Output: 6

Example 2: Input: arr = [1,1] Output: 1

Constraints:

1 <= arr.length <= 104 0 <= arr[i] <= 105

==
Problem Number: 236 URL: https://leetcode.com/problems/convert-binary-number-in-a-linked-list-to-integer Title: 1290. Convert Binary Number in a Linked List to Integer Problem Description: Given head which is a reference node to a singly-linked list. The value of each node in the linked list is either 0 or 1. The linked list holds the binary representation of a number. Return the decimal value of the number in the linked list. The most significant bit is at the head of the linked list. Example 1:

Input: head = [1,0,1] Output: 5 Explanation: (101) in base 2 = (5) in base 10

Example 2: Input: head = [0] Output: 0

Constraints:

The Linked List is not empty. Number of nodes will not exceed 30. Each node's value is either 0 or 1.

==
Problem Number: 237 URL: https://leetcode.com/problems/find-numbers-with-even-number-of-digits Title: 1295. Find Numbers with Even Number of Digits Problem Description: Given an array nums of integers, return how many of them contain an even number of digits. Example 1: Input: nums =

[12,345,2,6,7896] Output: 2 Explanation: 12 contains 2 digits (even number of digits). 345 contains 3 digits (odd number of digits). 2 contains 1 digit (odd number of digits). 6 contains 1 digit (odd number of digits). 7896 contains 4 digits (even number of digits). Therefore only 12 and 7896 contain an even number of digits.

Example 2: Input: nums = [555,901,482,1771] Output: 1 Explanation: Only 1771 contains an even number of digits.

Constraints:

1 <= nums.length <= 500 1 <= nums[i] <= 105

==
Problem Number: 238 URL: https://leetcode.com/problems/replace-elements-with-greatest-element-on-right-side Title: 1299. Replace Elements with Greatest Element on Right Side Problem Description: Given an array arr, replace every element in that array with the greatest element among the elements to its right, and replace the last element with -1. After doing so, return the array. Example 1: Input: arr = [17,18,5,4,6,1] Output: [18,6,6,6,1,-1] Explanation: - index 0 --> the greatest element to the right of index 0 is index 1 (18). - index 1 --> the greatest element to the right of index 1 is index 4 (6). - index 2 --> the greatest element to the right of index 2 is index 4 (6). - index 3 --> the greatest element to the right of index 3 is index 4 (6). - index 4 --> the greatest element to the right of index 4 is index 5 (1). - index 5 --> there are no elements to the right of index 5, so we put -1.

Example 2: Input: arr = [400] Output: [-1] Explanation: There are no elements to the right of index 0.

Constraints:

1 <= arr.length <= 104 1 <= arr[i] <= 105

==
Problem Number: 239 URL: https://leetcode.com/problems/find-n-unique-integers-sum-up-to-zero Title: 1304. Find N Unique Integers Sum up to Zero Problem Description: Given an integer n, return any array containing n unique integers such that they add up to 0. Example 1: Input: n = 5 Output: [-7,-1,1,3,4] Explanation: These arrays also are accepted [-5,-1,1,2,3] , [-3,-1,2,-2,4].

Example 2: Input: n = 3 Output: [-1,0,1]

Example 3: Input: n = 1 Output: [0]

Constraints:

1 <= n <= 1000

==
Problem Number: 240 URL: https://leetcode.com/problems/decrypt-string-

from-alphabet-to-integer-mapping Title: 1309. Decrypt String from Alphabet to Integer Mapping Problem Description: You are given a string s formed by digits and '#'. We want to map s to English lowercase characters as follows:

Characters ('a' to 'i') are represented by ('1' to '9') respectively. Characters ('j' to 'z') are represented by ('10#' to '26#') respectively.

Return the string formed after mapping. The test cases are generated so that a unique mapping will always exist.   Example 1: Input: s = "10#11#12" Output: "jkab" Explanation: "j" -> "10#" , "k" -> "11#" , "a" -> "1" , "b" -> "2".

Example 2: Input: s = "1326#" Output: "acz"

Constraints:

1 <= s.length <= 1000 s consists of digits and the '#' letter. s will be a valid string such that mapping is always possible.

==
Problem Number: 241 URL: https://leetcode.com/problems/decompress-run-length-encoded-list Title: 1313. Decompress Run-Length Encoded List Problem Description: We are given a list nums of integers representing a list compressed with run-length encoding. Consider each adjacent pair of elements [freq, val] = [nums[2*i], nums[2*i+1]] (with i >= 0). For each such pair, there are freq elements with value val concatenated in a sublist. Concatenate all the sublists from left to right to generate the decompressed list. Return the decompressed list.   Example 1: Input: nums = [1,2,3,4] Output: [2,4,4,4] Explanation: The first pair [1,2] means we have freq = 1 and val = 2 so we generate the array [2]. The second pair [3,4] means we have freq = 3 and val = 4 so we generate [4,4,4]. At the end the concatenation [2] + [4,4,4] is [2,4,4,4].

Example 2: Input: nums = [1,1,2,3] Output: [1,3,3]

Constraints:

2 <= nums.length <= 100 nums.length % 2 == 0 1 <= nums[i] <= 100

==
Problem Number: 242 URL: https://leetcode.com/problems/convert-integer-to-the-sum-of-two-no-zero-integers Title: 1317. Convert Integer to the Sum of Two No-Zero Integers Problem Description: No-Zero integer is a positive integer that does not contain any 0 in its decimal representation. Given an integer n, return a list of two integers [a, b] where:

a and b are No-Zero integers. a + b = n

The test cases are generated so that there is at least one valid solution. If there are many valid solutions, you can return any of them.   Example 1: Input: n = 2 Output: [1,1] Explanation: Let a = 1 and b = 1. Both a and b are no-zero integers, and a + b = 2 = n.

Example 2: Input: n = 11 Output: [2,9] Explanation: Let a = 2 and b = 9. Both a and b are no-zero integers, and a + b = 9 = n. Note that there are other valid answers as [8, 3] that can be accepted.

Constraints:

2 <= n <= 104

==

Problem Number: 243 URL: https://leetcode.com/problems/maximum-69-number Title: 1323. Maximum 69 Number Problem Description: You are given a positive integer num consisting only of digits 6 and 9. Return the maximum number you can get by changing at most one digit (6 becomes 9, and 9 becomes 6). Example 1: Input: num = 9669 Output: 9969 Explanation: Changing the first digit results in 6669. Changing the second digit results in 9969. Changing the third digit results in 9699. Changing the fourth digit results in 9666. The maximum number is 9969.

Example 2: Input: num = 9996 Output: 9999 Explanation: Changing the last digit 6 to 9 results in the maximum number.

Example 3: Input: num = 9999 Output: 9999 Explanation: It is better not to apply any change.

Constraints:

1 <= num <= 104 num consists of only 6 and 9 digits.

==

Problem Number: 244 URL: https://leetcode.com/problems/rank-transform-of-an-array Title: 1331. Rank Transform of an Array Problem Description: Given an array of integers arr, replace each element with its rank. The rank represents how large the element is. The rank has the following rules:

Rank is an integer starting from 1. The larger the element, the larger the rank. If two elements are equal, their rank must be the same. Rank should be as small as possible.

Example 1: Input: arr = [40,10,20,30] Output: [4,1,2,3] Explanation: 40 is the largest element. 10 is the smallest. 20 is the second smallest. 30 is the third smallest. Example 2: Input: arr = [100,100,100] Output: [1,1,1] Explanation: Same elements share the same rank.

Example 3: Input: arr = [37,12,28,9,100,56,80,5,12] Output: [5,3,4,2,8,6,7,1,3]

Constraints:

0 <= arr.length <= 105 -109 <= arr[i] <= 109

==

Problem Number: 245 URL: https://leetcode.com/problems/remove-palindromic-subsequences Title: 1332. Remove Palindromic Subsequences Problem Description: You are given a string s consisting only of letters 'a' and 'b'. In a single step you can remove one palindromic subsequence from s. Return the minimum number of steps to make the given string empty. A string is a subsequence of a given string if it is generated by deleting some characters of a given string without changing its order. Note that a subsequence does not necessarily need to be contiguous. A string is called palindrome if is one that reads the same backward as well as forward. Example 1: Input: s = "ababa" Output: 1 Explanation: s is already a palindrome, so its entirety can be removed in a single step.

Example 2: Input: s = "abb" Output: 2 Explanation: "abb" -> "bb" -> "". Remove palindromic subsequence "a" then "bb".

Example 3: Input: s = "baabb" Output: 2 Explanation: "baabb" -> "b" -> "". Remove palindromic subsequence "baab" then "b".

Constraints:

1 <= s.length <= 1000 s[i] is either 'a' or 'b'.

==

Problem Number: 246 URL: https://leetcode.com/problems/the-k-weakest-rows-in-a-matrix Title: 1337. The K Weakest Rows in a Matrix Problem Description: You are given an m x n binary matrix mat of 1's (representing soldiers) and 0's (representing civilians). The soldiers are positioned in front of the civilians. That is, all the 1's will appear to the left of all the 0's in each row. A row i is weaker than a row j if one of the following is true:

The number of soldiers in row i is less than the number of soldiers in row j. Both rows have the same number of soldiers and i < j.

Return the indices of the k weakest rows in the matrix ordered from weakest to strongest. Example 1: Input: mat = [[1,1,0,0,0], [1,1,1,1,0], [1,0,0,0,0], [1,1,0,0,0], [1,1,1,1,1]], k = 3 Output: [2,0,3] Explanation: The number of soldiers in each row is: - Row 0: 2 - Row 1: 4 - Row 2: 1 - Row 3: 2 - Row 4: 5 The rows ordered from weakest to strongest are [2,0,3,1,4].

Example 2: Input: mat = [[1,0,0,0], [1,1,1,1], [1,0,0,0], [1,0,0,0]], k = 2 Output: [0,2] Explanation: The number of soldiers in each row is: - Row 0: 1 - Row 1: 4 - Row 2: 1 - Row 3: 1 The rows ordered from weakest to strongest are [0,2,3,1].

Constraints:

m == mat.length n == mat[i].length 2 <= n, m <= 100 1 <= k <= m matrix[i][j] is either 0 or 1.

==

Problem Number: 247 URL: https://leetcode.com/problems/number-of-steps-to-reduce-a-number-to-zero Title: 1342. Number of Steps to Reduce a Number to Zero Problem Description: Given an integer num, return the number of steps to reduce it to zero. In one step, if the current number is even, you have to divide it by 2, otherwise, you have to subtract 1 from it. Example 1: Input:

num = 14 Output: 6 Explanation: Step 1) 14 is even; divide by 2 and obtain 7. Step 2) 7 is odd; subtract 1 and obtain 6. Step 3) 6 is even; divide by 2 and obtain 3. Step 4) 3 is odd; subtract 1 and obtain 2. Step 5) 2 is even; divide by 2 and obtain 1. Step 6) 1 is odd; subtract 1 and obtain 0.

Example 2: Input: num = 8 Output: 4 Explanation: Step 1) 8 is even; divide by 2 and obtain 4. Step 2) 4 is even; divide by 2 and obtain 2. Step 3) 2 is even; divide by 2 and obtain 1. Step 4) 1 is odd; subtract 1 and obtain 0.

Example 3: Input: num = 123 Output: 12

Constraints:

0 <= num <= 106

==
Problem Number: 248 URL: https://leetcode.com/problems/check-if-n-and-its-double-exist Title: 1346. Check If N and Its Double Exist Problem Description: Given an array arr of integers, check if there exist two indices i and j such that :

i != j 0 <= i, j < arr.length arr[i] == 2 * arr[j]

Example 1: Input: arr = [10,2,5,3] Output: true Explanation: For i = 0 and j = 2, arr[i] == 10 == 2 * 5 == 2 * arr[j]

Example 2: Input: arr = [3,1,7,11] Output: false Explanation: There is no i and j that satisfy the conditions.

Constraints:

2 <= arr.length <= 500 -103 <= arr[i] <= 103

==
Problem Number: 249 URL: https://leetcode.com/problems/count-negative-numbers-in-a-sorted-matrix Title: 1351. Count Negative Numbers in a Sorted Matrix Problem Description: Given a m x n matrix grid which is sorted in non-increasing order both row-wise and column-wise, return the number of negative numbers in grid. Example 1: Input: grid = [[4,3,2,-1],[3,2,1,-1],[1,1,-1,-2],[-1,-1,-2,-3]] Output: 8 Explanation: There are 8 negatives number in the matrix.

Example 2: Input: grid = [[3,2],[1,0]] Output: 0

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 100 -100 <= grid[i][j] <= 100

Follow up: Could you find an O(n + m) solution? ==
Problem Number: 250 URL: https://leetcode.com/problems/sort-integers-by-the-number-of-1-bits Title: 1356. Sort Integers by The Number of 1 Bits Problem Description: You are given an integer array arr. Sort the integers in

the array in ascending order by the number of 1's in their binary representation and in case of two or more integers have the same number of 1's you have to sort them in ascending order. Return the array after sorting it. Example 1: Input: arr = [0,1,2,3,4,5,6,7,8] Output: [0,1,2,4,8,3,5,6,7] Explantion: [0] is the only integer with 0 bits. [1,2,4,8] all have 1 bit. [3,5,6] have 2 bits. [7] has 3 bits. The sorted array by bits is [0,1,2,4,8,3,5,6,7]

Example 2: Input: arr = [1024,512,256,128,64,32,16,8,4,2,1] Output: [1,2,4,8,16,32,64,128,256,512,1024] Explantion: All integers have 1 bit in the binary representation, you should just sort them in ascending order.

Constraints:

1 <= arr.length <= 500 0 <= arr[i] <= 104

==
Problem Number: 251 URL: https://leetcode.com/problems/number-of-days-between-two-dates Title: 1360. Number of Days Between Two Dates Problem Description: Write a program to count the number of days between two dates. The two dates are given as strings, their format is YYYY-MM-DD as shown in the examples. Example 1: Input: date1 = "2019-06-29", date2 = "2019-06-30" Output: 1 Example 2: Input: date1 = "2020-01-15", date2 = "2019-12-31" Output: 15

Constraints:

The given dates are valid dates between the years 1971 and 2100.

==
Problem Number: 252 URL: https://leetcode.com/problems/how-many-numbers-are-smaller-than-the-current-number Title: 1365. How Many Numbers Are Smaller Than the Current Number Problem Description: Given the array nums, for each nums[i] find out how many numbers in the array are smaller than it. That is, for each nums[i] you have to count the number of valid j's such that j != i and nums[j] < nums[i]. Return the answer in an array. Example 1: Input: nums = [8,1,2,2,3] Output: [4,0,1,1,3] Explanation: For nums[0]=8 there exist four smaller numbers than it (1, 2, 2 and 3). For nums[1]=1 does not exist any smaller number than it. For nums[2]=2 there exist one smaller number than it (1). For nums[3]=2 there exist one smaller number than it (1). For nums[4]=3 there exist three smaller numbers than it (1, 2 and 2).

Example 2: Input: nums = [6,5,4,8] Output: [2,1,0,3]

Example 3: Input: nums = [7,7,7,7] Output: [0,0,0,0]

Constraints:

2 <= nums.length <= 500 0 <= nums[i] <= 100

==
Problem Number: 253 URL: https://leetcode.com/problems/increasing-

decreasing-string Title: 1370. Increasing Decreasing String Problem Description: You are given a string s. Reorder the string using the following algorithm:

Pick the smallest character from s and append it to the result. Pick the smallest character from s which is greater than the last appended character to the result and append it. Repeat step 2 until you cannot pick more characters. Pick the largest character from s and append it to the result. Pick the largest character from s which is smaller than the last appended character to the result and append it. Repeat step 5 until you cannot pick more characters. Repeat the steps from 1 to 6 until you pick all characters from s.

In each step, If the smallest or the largest character appears more than once you can choose any occurrence and append it to the result. Return the result string after sorting s with this algorithm. Example 1: Input: s = "aaaabbbbcccc" Output: "abccbaabccba" Explanation: After steps 1, 2 and 3 of the first iteration, result = "abc" After steps 4, 5 and 6 of the first iteration, result = "abccba" First iteration is done. Now s = "aabbcc" and we go back to step 1 After steps 1, 2 and 3 of the second iteration, result = "abccbaabc" After steps 4, 5 and 6 of the second iteration, result = "abccbaabccba"

Example 2: Input: s = "rat" Output: "art" Explanation: The word "rat" becomes "art" after re-ordering it with the mentioned algorithm.

Constraints:

1 <= s.length <= 500 s consists of only lowercase English letters.

==

Problem Number: 254 URL: https://leetcode.com/problems/generate-a-string-with-characters-that-have-odd-counts Title: 1374. Generate a String With Characters That Have Odd Counts Problem Description: Given an integer n, return a string with n characters such that each character in such string occurs an odd number of times. The returned string must contain only lowercase English letters. If there are multiples valid strings, return any of them. Example 1: Input: n = 4 Output: "pppz" Explanation: "pppz" is a valid string since the character 'p' occurs three times and the character 'z' occurs once. Note that there are many other valid strings such as "ohhh" and "love".

Example 2: Input: n = 2 Output: "xy" Explanation: "xy" is a valid string since the characters 'x' and 'y' occur once. Note that there are many other valid strings such as "ag" and "ur".

Example 3: Input: n = 7 Output: "holasss"

Constraints:

1 <= n <= 500

==

Problem Number: 255 URL: https://leetcode.com/problems/find-a-corresponding-node-of-a-binary-tree-in-a-clone-of-that-tree Title: 1379. Find a Corresponding Node of a Binary Tree in a Clone of That Tree Problem Description: Given two binary trees original and cloned and given a reference to a node target in the original tree. The cloned tree is a copy of the original tree. Return a reference to the same node in the cloned tree. Note that you are not allowed to change any of the two trees or the target node and the answer must be a reference to a node in the cloned tree.   Example 1:

Input: tree = [7,4,3,null,null,6,19], target = 3 Output: 3 Explanation: In all examples the original and cloned trees are shown. The target node is a green node from the original tree. The answer is the yellow node from the cloned tree.

Example 2:

Input: tree = [7], target = 7 Output: 7

Example 3:

Input: tree = [8,null,6,null,5,null,4,null,3,null,2,null,1], target = 4 Output: 4

Constraints:

The number of nodes in the tree is in the range [1, 104]. The values of the nodes of the tree are unique. target node is a node from the original tree and is not null.

Follow up: Could you solve the problem if repeated values on the tree are allowed?

==

Problem Number: 256 URL: https://leetcode.com/problems/lucky-numbers-in-a-matrix Title: 1380. Lucky Numbers in a Matrix Problem Description: Given an m x n matrix of distinct numbers, return all lucky numbers in the matrix in any order. A lucky number is an element of the matrix such that it is the minimum element in its row and maximum in its column.   Example 1: Input: matrix = [[3,7,8],[9,11,13],[15,16,17]] Output: [15] Explanation: 15 is the only lucky number since it is the minimum in its row and the maximum in its column.

Example 2: Input: matrix = [[1,10,4,2],[9,3,8,7],[15,16,17,12]] Output: [12] Explanation: 12 is the only lucky number since it is the minimum in its row and the maximum in its column.

Example 3: Input: matrix = [[7,8],[1,2]] Output: [7] Explanation: 7 is the only lucky number since it is the minimum in its row and the maximum in its column.

Constraints:

m == mat.length n == mat[i].length 1 <= n, m <= 50 1 <= matrix[i][j] <= 105. All elements in the matrix are distinct.

==

Problem Number: 257 URL: https://leetcode.com/problems/find-the-distance-value-between-two-arrays Title: 1385. Find the Distance Value Between Two Arrays Problem Description: Given two integer arrays arr1 and arr2, and the integer d, return the distance value between the two arrays. The distance value is defined as the number of elements arr1[i] such that there is not any element arr2[j] where |arr1[i]-arr2[j]| <= d.   Example 1: Input: arr1 = [4,5,8], arr2 = [10,9,1,8], d = 2 Output: 2 Explanation: For arr1[0]=4 we have: |4-10|=6 > d=2 |4-9|=5 > d=2 |4-1|=3 > d=2 |4-8|=4 > d=2 For arr1[1]=5 we have: |5-10|=5 > d=2 |5-9|=4 > d=2 |5-1|=4 > d=2 |5-8|=3 > d=2 For arr1[2]=8 we have: |8-10|=2 <= d=2 |8-9|=1 <= d=2 |8-1|=7 > d=2 |8-8|=0 <= d=2

Example 2: Input: arr1 = [1,4,2,3], arr2 = [-4,-3,6,10,20,30], d = 3 Output: 2

Example 3: Input: arr1 = [2,1,100,3], arr2 = [-5,-2,10,-3,7], d = 6 Output: 1

Constraints:

1 <= arr1.length, arr2.length <= 500 -1000 <= arr1[i], arr2[j] <= 1000 0 <= d <= 100

==

Problem Number: 258 URL: https://leetcode.com/problems/create-target-array-in-the-given-order Title: 1389. Create Target Array in the Given Order Problem Description: Given two arrays of integers nums and index. Your task is to create target array under the following rules:

Initially target array is empty. From left to right read nums[i] and index[i], insert at index index[i] the value nums[i] in target array. Repeat the previous step until there are no elements to read in nums and index.

Return the target array. It is guaranteed that the insertion operations will be valid.   Example 1: Input: nums = [0,1,2,3,4], index = [0,1,2,2,1] Output: [0,4,1,3,2] Explanation: nums index target 0 0 [0] 1 1 [0,1] 2 2 [0,1,2] 3 2 [0,1,3,2] 4 1 [0,4,1,3,2]

Example 2: Input: nums = [1,2,3,4,0], index = [0,1,2,3,0] Output: [0,1,2,3,4] Explanation: nums index target 1 0 [1] 2 1 [1,2] 3 2 [1,2,3] 4 3 [1,2,3,4] 0 0 [0,1,2,3,4]

Example 3: Input: nums = [1], index = [0] Output: [1]

Constraints:

1 <= nums.length, index.length <= 100 nums.length == index.length 0 <= nums[i] <= 100 0 <= index[i] <= i

==

Problem Number: 259 URL: https://leetcode.com/problems/find-lucky-integer-in-an-array Title: 1394. Find Lucky Integer in an Array Problem Description: Given an array of integers arr, a lucky integer is an integer that has a frequency in the array equal to its value. Return the largest lucky integer in the array. If there is no lucky integer return -1. Example 1: Input: arr = [2,2,3,4] Output: 2 Explanation: The only lucky number in the array is 2 because frequency[2] == 2.

Example 2: Input: arr = [1,2,2,3,3,3] Output: 3 Explanation: 1, 2 and 3 are all lucky numbers, return the largest of them.

Example 3: Input: arr = [2,2,2,3,3] Output: -1 Explanation: There are no lucky numbers in the array.

Constraints:

1 <= arr.length <= 500 1 <= arr[i] <= 500

==

Problem Number: 260 URL: https://leetcode.com/problems/count-largest-group Title: 1399. Count Largest Group Problem Description: You are given an integer n. Each number from 1 to n is grouped according to the sum of its digits. Return the number of groups that have the largest size. Example 1: Input: n = 13 Output: 4 Explanation: There are 9 groups in total, they are grouped according sum of its digits of numbers from 1 to 13: [1,10], [2,11], [3,12], [4,13], [5], [6], [7], [8], [9]. There are 4 groups with largest size.

Example 2: Input: n = 2 Output: 2 Explanation: There are 2 groups [1], [2] of size 1.

Constraints:

1 <= n <= 104

==

Problem Number: 261 URL: https://leetcode.com/problems/minimum-subsequence-in-non-increasing-order Title: 1403. Minimum Subsequence in Non-Increasing Order Problem Description: Given the array nums, obtain a subsequence of the array whose sum of elements is strictly greater than the sum of the non included elements in such subsequence. If there are multiple solutions, return the subsequence with minimum size and if there still exist multiple solutions, return the subsequence with the maximum total sum of all its elements. A subsequence of an array can be obtained by erasing some (possibly zero) elements from the array. Note that the solution with the given constraints is guaranteed to be unique. Also return the answer sorted in non-increasing order. Example 1: Input: nums = [4,3,10,9,8] Output: [10,9] Explanation: The subsequences [10,9] and [10,8] are minimal such that the sum of their elements is strictly greater than the sum of elements not included. However, the subsequence [10,9] has the maximum total sum of its elements.

Example 2: Input: nums = [4,4,7,6,7] Output: [7,7,6] Explanation: The subsequence [7,7] has the sum of its elements equal to 14 which is not strictly greater than the sum of elements not included (14 = 4 + 4 + 6). Therefore, the subsequence [7,6,7] is the minimal satisfying the conditions. Note the subsequence has to be returned in non-decreasing order.

Constraints:

1 <= nums.length <= 500 1 <= nums[i] <= 100

==
Problem Number: 262 URL: https://leetcode.com/problems/string-matching-in-an-array Title: 1408. String Matching in an Array Problem Description: Given an array of string words, return all strings in words that is a substring of another word. You can return the answer in any order. A substring is a contiguous sequence of characters within a string Example 1: Input: words = ["mass","as","hero","superhero"] Output: ["as","hero"] Explanation: "as" is substring of "mass" and "hero" is substring of "superhero". ["hero","as"] is also a valid answer.

Example 2: Input: words = ["leetcode","et","code"] Output: ["et","code"] Explanation: "et", "code" are substring of "leetcode".

Example 3: Input: words = ["blue","green","bu"] Output: [] Explanation: No string of words is substring of another string.

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 30 words[i] contains only lowercase English letters. All the strings of words are unique.

==
Problem Number: 263 URL: https://leetcode.com/problems/minimum-value-to-get-positive-step-by-step-sum Title: 1413. Minimum Value to Get Positive Step by Step Sum Problem Description: Given an array of integers nums, you start with an initial positive value startValue. In each iteration, you calculate the step by step sum of startValue plus elements in nums (from left to right). Return the minimum positive value of startValue such that the step by step sum is never less than 1. Example 1: Input: nums = [-3,2,-3,4,2] Output: 5 Explanation: If you choose startValue = 4, in the third iteration your step by step sum is less than 1. step by step sum startValue = 4 | startValue = 5 | nums (4 -3) = 1 | (5 -3) = 2 | -3 (1 +2) = 3 | (2 +2) = 4 | 2 (3 -3) = 0 | (4 -3) = 1 | -3 (0 +4) = 4 | (1 +4) = 5 | 4 (4 +2) = 6 | (5 +2) = 7 | 2

Example 2: Input: nums = [1,2] Output: 1 Explanation: Minimum start value should be positive.

Example 3: Input: nums = [1,-2,-3] Output: 5

Constraints:

1 <= nums.length <= 100 -100 <= nums[i] <= 100

==
Problem Number: 264 URL: https://leetcode.com/problems/reformat-the-string Title: 1417. Reformat The String Problem Description: You are given an alphanumeric string s. (Alphanumeric string is a string consisting of lowercase English letters and digits). You have to find a permutation of the string where

no letter is followed by another letter and no digit is followed by another digit. That is, no two adjacent characters have the same type. Return the reformatted string or return an empty string if it is impossible to reformat the string. Example 1: Input: s = "a0b1c2" Output: "0a1b2c" Explanation: No two adjacent characters have the same type in "0a1b2c". "a0b1c2", "0a1b2c", "0c2a1b" are also valid permutations.

Example 2: Input: s = "leetcode" Output: "" Explanation: "leetcode" has only characters so we cannot separate them by digits.

Example 3: Input: s = "1229857369" Output: "" Explanation: "1229857369" has only digits so we cannot separate them by characters.

Constraints:

1 <= s.length <= 500 s consists of only lowercase English letters and/or digits.

==
Problem Number: 265 URL: https://leetcode.com/problems/maximum-score-after-splitting-a-string Title: 1422. Maximum Score After Splitting a String Problem Description: Given a string s of zeros and ones, return the maximum score after splitting the string into two non-empty substrings (i.e. left substring and right substring). The score after splitting a string is the number of zeros in the left substring plus the number of ones in the right substring. Example 1: Input: s = "011101" Output: 5 Explanation: All possible ways of splitting s into two non-empty substrings are: left = "0" and right = "11101", score = 1 + 4 = 5 left = "01" and right = "1101", score = 1 + 3 = 4 left = "011" and right = "101", score = 1 + 2 = 3 left = "0111" and right = "01", score = 1 + 1 = 2 left = "01110" and right = "1", score = 2 + 1 = 3

Example 2: Input: s = "00111" Output: 5 Explanation: When left = "00" and right = "111", we get the maximum score = 2 + 3 = 5

Example 3: Input: s = "1111" Output: 3

Constraints:

2 <= s.length <= 500 The string s consists of characters '0' and '1' only.

==
Problem Number: 266 URL: https://leetcode.com/problems/kids-with-the-greatest-number-of-candies Title: 1431. Kids With the Greatest Number of Candies Problem Description: There are n kids with candies. You are given an integer array candies, where each candies[i] represents the number of candies the ith kid has, and an integer extraCandies, denoting the number of extra candies that you have. Return a boolean array result of length n, where result[i] is true if, after giving the ith kid all the extraCandies, they will have the greatest number of candies among all the kids, or false otherwise. Note that multiple kids can have the greatest number of candies. Example 1: Input: candies = [2,3,5,1,3], extraCandies = 3 Output: [true,true,true,false,true] Explanation: If you give all extraCandies to: - Kid 1, they will have 2 + 3 =

5 candies, which is the greatest among the kids. - Kid 2, they will have 3 + 3 = 6 candies, which is the greatest among the kids. - Kid 3, they will have 5 + 3 = 8 candies, which is the greatest among the kids. - Kid 4, they will have 1 + 3 = 4 candies, which is not the greatest among the kids. - Kid 5, they will have 3 + 3 = 6 candies, which is the greatest among the kids.

Example 2: Input: candies = [4,2,1,1,2], extraCandies = 1 Output: [true,false,false,false,false] Explanation: There is only 1 extra candy. Kid 1 will always have the greatest number of candies, even if a different kid is given the extra candy.

Example 3: Input: candies = [12,1,12], extraCandies = 10 Output: [true,false,true]

Constraints:

n == candies.length 2 <= n <= 100 1 <= candies[i] <= 100 1 <= extraCandies <= 50

==

Problem Number: 267 URL: https://leetcode.com/problems/destination-city Title: 1436. Destination City Problem Description: You are given the array paths, where paths[i] = [cityAi, cityBi] means there exists a direct path going from cityAi to cityBi. Return the destination city, that is, the city without any path outgoing to another city. It is guaranteed that the graph of paths forms a line without any loop, therefore, there will be exactly one destination city. Example 1: Input: paths = [[”London”,”New York”],[”New York”,”Lima”],[”Lima”,”Sao Paulo”]] Output: ”Sao Paulo” Explanation: Starting at ”London” city you will reach ”Sao Paulo” city which is the destination city. Your trip consist of: ”London” -> ”New York” -> ”Lima” -> ”Sao Paulo”.

Example 2: Input: paths = [[”B”,”C”],[”D”,”B”],[”C”,”A”]] Output: ”A” Explanation: All possible trips are: ”D” -> ”B” -> ”C” -> ”A”. ”B” -> ”C” -> ”A”. ”C” -> ”A”. ”A”. Clearly the destination city is ”A”.

Example 3: Input: paths = [[”A”,”Z”]] Output: ”Z”

Constraints:

1 <= paths.length <= 100 paths[i].length == 2 1 <= cityAi.length, cityBi.length <= 10 cityAi != cityBi All strings consist of lowercase and uppercase English letters and the space character.

==

Problem Number: 268 URL: https://leetcode.com/problems/check-if-all-1s-are-at-least-length-k-places-away Title: 1437. Check If All 1’s Are at Least Length K Places Away Problem Description: Given an binary array nums and an integer k, return true if all 1’s are at least k places away from each other, otherwise return false. Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2 Output: true Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2 Output: false Explanation: The second 1 and third 1 are only one apart from each other.

Constraints:

1 <= nums.length <= 105 0 <= k <= nums.length nums[i] is 0 or 1

==
Problem Number: 269 URL: https://leetcode.com/problems/consecutive-characters Title: 1446. Consecutive Characters Problem Description: The power of the string is the maximum length of a non-empty substring that contains only one unique character. Given a string s, return the power of s. Example 1: Input: s = "leetcode" Output: 2 Explanation: The substring "ee" is of length 2 with the character 'e' only.

Example 2: Input: s = "abbcccddddeeeeedcba" Output: 5 Explanation: The substring "eeeee" is of length 5 with the character 'e' only.

Constraints:

1 <= s.length <= 500 s consists of only lowercase English letters.

==
Problem Number: 270 URL: https://leetcode.com/problems/number-of-students-doing-homework-at-a-given-time Title: 1450. Number of Students Doing Homework at a Given Time Problem Description: Given two integer arrays startTime and endTime and given an integer queryTime. The ith student started doing their homework at the time startTime[i] and finished it at time endTime[i]. Return the number of students doing their homework at time queryTime. More formally, return the number of students where queryTime lays in the interval [startTime[i], endTime[i]] inclusive. Example 1: Input: startTime = [1,2,3], endTime = [3,2,7], queryTime = 4 Output: 1 Explanation: We have 3 students where: The first student started doing homework at time 1 and finished at time 3 and wasn't doing anything at time 4. The second student started doing homework at time 2 and finished at time 2 and also wasn't doing anything at time 4. The third student started doing homework at time 3 and finished at time 7 and was the only student doing homework at time 4.

Example 2: Input: startTime = [4], endTime = [4], queryTime = 4 Output: 1 Explanation: The only student was doing their homework at the queryTime.

Constraints:

startTime.length == endTime.length 1 <= startTime.length <= 100 1 <= startTime[i] <= endTime[i] <= 1000 1 <= queryTime <= 1000

==

Problem Number: 271 URL: https://leetcode.com/problems/check-if-a-word-occurs-as-a-prefix-of-any-word-in-a-sentence Title: 1455. Check If a Word Occurs As a Prefix of Any Word in a Sentence Problem Description: Given a sentence that consists of some words separated by a single space, and a searchWord, check if searchWord is a prefix of any word in sentence. Return the index of the word in sentence (1-indexed) where searchWord is a prefix of this word. If searchWord is a prefix of more than one word, return the index of the first word (minimum index). If there is no such word return -1. A prefix of a string s is any leading contiguous substring of s. Example 1: Input: sentence = "i love eating burger", searchWord = "burg" Output: 4 Explanation: "burg" is prefix of "burger" which is the 4th word in the sentence.

Example 2: Input: sentence = "this problem is an easy problem", searchWord = "pro" Output: 2 Explanation: "pro" is prefix of "problem" which is the 2nd and the 6th word in the sentence, but we return 2 as it's the minimal index.

Example 3: Input: sentence = "i am tired", searchWord = "you" Output: -1 Explanation: "you" is not a prefix of any word in the sentence.

Constraints:

1 <= sentence.length <= 100 1 <= searchWord.length <= 10 sentence consists of lowercase English letters and spaces. searchWord consists of lowercase English letters.

==

Problem Number: 272 URL: https://leetcode.com/problems/make-two-arrays-equal-by-reversing-subarrays Title: 1460. Make Two Arrays Equal by Reversing Subarrays Problem Description: You are given two integer arrays of equal length target and arr. In one step, you can select any non-empty subarray of arr and reverse it. You are allowed to make any number of steps. Return true if you can make arr equal to target or false otherwise. Example 1: Input: target = [1,2,3,4], arr = [2,4,1,3] Output: true Explanation: You can follow the next steps to convert arr to target: 1- Reverse subarray [2,4,1], arr becomes [1,4,2,3] 2- Reverse subarray [4,2], arr becomes [1,2,4,3] 3- Reverse subarray [4,3], arr becomes [1,2,3,4] There are multiple ways to convert arr to target, this is not the only way to do so.

Example 2: Input: target = [7], arr = [7] Output: true Explanation: arr is equal to target without any reverses.

Example 3: Input: target = [3,7,9], arr = [3,7,11] Output: false Explanation: arr does not have value 9 and it can never be converted to target.

Constraints:

target.length == arr.length 1 <= target.length <= 1000 1 <= target[i] <= 1000 1 <= arr[i] <= 1000

==
Problem Number: 273 URL: https://leetcode.com/problems/maximum-product-of-two-elements-in-an-array Title: 1464. Maximum Product of Two Elements in an Array Problem Description: Given the array of integers nums, you will choose two different indices i and j of that array. Return the maximum value of (nums[i]-1)*(nums[j]-1). Example 1: Input: nums = [3,4,5,2] Output: 12 Explanation: If you choose the indices i=1 and j=2 (indexed from 0), you will get the maximum value, that is, (nums[1]-1)*(nums[2]-1) = (4-1)*(5-1) = 3*4 = 12.

Example 2: Input: nums = [1,5,4,5] Output: 16 Explanation: Choosing the indices i=1 and j=3 (indexed from 0), you will get the maximum value of (5-1)*(5-1) = 16.

Example 3: Input: nums = [3,7] Output: 12

Constraints:

2 <= nums.length <= 500 1 <= nums[i] <= 10^3
==
Problem Number: 274 URL: https://leetcode.com/problems/shuffle-the-array Title: 1470. Shuffle the Array Problem Description: Given the array nums consisting of 2n elements in the form [x1,x2,...,xn,y1,y2,...,yn]. Return the array in the form [x1,y1,x2,y2,...,xn,yn]. Example 1: Input: nums = [2,5,1,3,4,7], n = 3 Output: [2,3,5,4,1,7] Explanation: Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is [2,3,5,4,1,7].

Example 2: Input: nums = [1,2,3,4,4,3,2,1], n = 4 Output: [1,4,2,3,3,2,4,1]

Example 3: Input: nums = [1,1,2,2], n = 2 Output: [1,2,1,2]

Constraints:

1 <= n <= 500 nums.length == 2n 1 <= nums[i] <= 10^3
==
Problem Number: 275 URL: https://leetcode.com/problems/final-prices-with-a-special-discount-in-a-shop Title: 1475. Final Prices With a Special Discount in a Shop Problem Description: You are given an integer array prices where prices[i] is the price of the ith item in a shop. There is a special discount for items in the shop. If you buy the ith item, then you will receive a discount equivalent to prices[j] where j is the minimum index such that j > i and prices[j] <= prices[i]. Otherwise, you will not receive any discount at all. Return an integer array answer where answer[i] is the final price you will pay for the ith item of the shop, considering the special discount. Example 1: Input: prices = [8,4,6,2,3] Output: [4,2,4,2,3] Explanation: For item 0 with price[0]=8 you will receive a discount equivalent to prices[1]=4, therefore, the final price you will pay is 8 - 4 = 4. For item 1 with price[1]=4 you will receive a discount equivalent to prices[3]=2, therefore, the final price you will pay is 4 - 2 = 2. For item 2 with price[2]=6 you will receive a discount equivalent to prices[3]=2,

therefore, the final price you will pay is 6 - 2 = 4. For items 3 and 4 you will not receive any discount at all.

Example 2: Input: prices = [1,2,3,4,5] Output: [1,2,3,4,5] Explanation: In this case, for all items, you will not receive any discount at all.

Example 3: Input: prices = [10,1,1,6] Output: [9,0,1,6]

Constraints:

1 <= prices.length <= 500 1 <= prices[i] <= 1000

==
Problem Number: 276 URL: https://leetcode.com/problems/running-sum-of-1d-array Title: 1480. Running Sum of 1d Array Problem Description: Given an array nums. We define a running sum of an array as runningSum[i] = sum(nums[0]…nums[i]). Return the running sum of nums. Example 1: Input: nums = [1,2,3,4] Output: [1,3,6,10] Explanation: Running sum is obtained as follows: [1, 1+2, 1+2+3, 1+2+3+4]. Example 2: Input: nums = [1,1,1,1,1] Output: [1,2,3,4,5] Explanation: Running sum is obtained as follows: [1, 1+1, 1+1+1, 1+1+1+1, 1+1+1+1+1]. Example 3: Input: nums = [3,1,2,10,1] Output: [3,4,6,16,17]

Constraints:

1 <= nums.length <= 1000 -10^6 <= nums[i] <= 10^6

==
Problem Number: 277 URL: https://leetcode.com/problems/xor-operation-in-an-array Title: 1486. XOR Operation in an Array Problem Description: You are given an integer n and an integer start. Define an array nums where nums[i] = start + 2 * i (0-indexed) and n == nums.length. Return the bitwise XOR of all elements of nums. Example 1: Input: n = 5, start = 0 Output: 8 Explanation: Array nums is equal to [0, 2, 4, 6, 8] where (0 ^ 2 ^ 4 ^ 6 ^ 8) = 8. Where "^" corresponds to bitwise XOR operator.

Example 2: Input: n = 4, start = 3 Output: 8 Explanation: Array nums is equal to [3, 5, 7, 9] where (3 ^ 5 ^ 7 ^ 9) = 8.

Constraints:

1 <= n <= 1000 0 <= start <= 1000 n == nums.length

==
Problem Number: 278 URL: https://leetcode.com/problems/average-salary-excluding-the-minimum-and-maximum-salary Title: 1491. Average Salary Excluding the Minimum and Maximum Salary Problem Description: You are given an array of unique integers salary where salary[i] is the salary of the ith employee. Return the average salary of employees excluding the minimum and maximum salary. Answers within 10-5 of the actual answer will be accepted. Example 1: Input: salary = [4000,3000,1000,2000] Output: 2500.00000 Explanation: Minimum salary and maximum salary are 1000 and

4000 respectively. Average salary excluding minimum and maximum salary is (2000+3000) / 2 = 2500

Example 2: Input: salary = [1000,2000,3000] Output: 2000.00000 Explanation: Minimum salary and maximum salary are 1000 and 3000 respectively. Average salary excluding minimum and maximum salary is (2000) / 1 = 2000

Constraints:

3 <= salary.length <= 100 1000 <= salary[i] <= 106 All the integers of salary are unique.

==

Problem Number: 279 URL: https://leetcode.com/problems/path-crossing Title: 1496. Path Crossing Problem Description: Given a string path, where path[i] = 'N', 'S', 'E' or 'W', each representing moving one unit north, south, east, or west, respectively. You start at the origin (0, 0) on a 2D plane and walk on the path specified by path. Return true if the path crosses itself at any point, that is, if at any time you are on a location you have previously visited. Return false otherwise. Example 1:

Input: path = "NES" Output: false Explanation: Notice that the path doesn't cross any point more than once.

Example 2:

Input: path = "NESWW" Output: true Explanation: Notice that the path visits the origin twice. Constraints:

1 <= path.length <= 104 path[i] is either 'N', 'S', 'E', or 'W'.

==

Problem Number: 280 URL: https://leetcode.com/problems/can-make-arithmetic-progression-from-sequence Title: 1502. Can Make Arithmetic Progression From Sequence Problem Description: A sequence of numbers is called an arithmetic progression if the difference between any two consecutive elements is the same. Given an array of numbers arr, return true if the array can be rearranged to form an arithmetic progression. Otherwise, return false. Example 1: Input: arr = [3,5,1] Output: true Explanation: We can reorder the elements as [1,3,5] or [5,3,1] with differences 2 and -2 respectively, between each consecutive elements.

Example 2: Input: arr = [1,2,4] Output: false Explanation: There is no way to reorder the elements to obtain an arithmetic progression.

Constraints:

2 <= arr.length <= 1000 -106 <= arr[i] <= 106

==

Problem Number: 281 URL: https://leetcode.com/problems/reformat-date

Title: 1507. Reformat Date Problem Description: Given a date string in the form Day Month Year, where:

Day is in the set {"1st", "2nd", "3rd", "4th", ..., "30th", "31st"}. Month is in the set {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"}. Year is in the range [1900, 2100].

Convert the date string to the format YYYY-MM-DD, where:

YYYY denotes the 4 digit year. MM denotes the 2 digit month. DD denotes the 2 digit day.

Example 1: Input: date = "20th Oct 2052" Output: "2052-10-20"

Example 2: Input: date = "6th Jun 1933" Output: "1933-06-06"

Example 3: Input: date = "26th May 1960" Output: "1960-05-26"

Constraints:

The given dates are guaranteed to be valid, so no error handling is necessary.

==
Problem Number: 282 URL: https://leetcode.com/problems/number-of-good-pairs Title: 1512. Number of Good Pairs Problem Description: Given an array of integers nums, return the number of good pairs. A pair (i, j) is called good if nums[i] == nums[j] and i < j. Example 1: Input: nums = [1,2,3,1,1,3] Output: 4 Explanation: There are 4 good pairs (0,3), (0,4), (3,4), (2,5) 0-indexed.

Example 2: Input: nums = [1,1,1,1] Output: 6 Explanation: Each pair in the array are good.

Example 3: Input: nums = [1,2,3] Output: 0

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100

==
Problem Number: 283 URL: https://leetcode.com/problems/water-bottles Title: 1518. Water Bottles Problem Description: There are numBottles water bottles that are initially full of water. You can exchange numExchange empty water bottles from the market with one full water bottle. The operation of drinking a full water bottle turns it into an empty bottle. Given the two integers numBottles and numExchange, return the maximum number of water bottles you can drink. Example 1:

Input: numBottles = 9, numExchange = 3 Output: 13 Explanation: You can exchange 3 empty bottles to get 1 full water bottle. Number of water bottles you can drink: 9 + 3 + 1 = 13.

Example 2:

Input: numBottles = 15, numExchange = 4 Output: 19 Explanation: You can exchange 4 empty bottles to get 1 full water bottle. Number of water bottles you can drink: 15 + 3 + 1 = 19.

Constraints:

1 <= numBottles <= 100 2 <= numExchange <= 100

==
Problem Number: 284 URL: https://leetcode.com/problems/count-odd-numbers-in-an-interval-range Title: 1523. Count Odd Numbers in an Interval Range Problem Description: Given two non-negative integers low and high. Return the count of odd numbers between low and high (inclusive). Example 1: Input: low = 3, high = 7 Output: 3 Explanation: The odd numbers between 3 and 7 are [3,5,7]. Example 2: Input: low = 8, high = 10 Output: 1 Explanation: The odd numbers between 8 and 10 are [9]. Constraints:

0 <= low <= high <= 10^9

==
Problem Number: 285 URL: https://leetcode.com/problems/shuffle-string Title: 1528. Shuffle String Problem Description: You are given a string s and an integer array indices of the same length. The string s will be shuffled such that the character at the ith position moves to indices[i] in the shuffled string. Return the shuffled string. Example 1:

Input: s = "codeleet", indices = [4,5,6,7,0,2,1,3] Output: "leetcode" Explanation: As shown, "codeleet" becomes "leetcode" after shuffling.

Example 2: Input: s = "abc", indices = [0,1,2] Output: "abc" Explanation: After shuffling, each character remains in its position.

Constraints:

s.length == indices.length == n 1 <= n <= 100 s consists of only lowercase English letters. 0 <= indices[i] < n All values of indices are unique.

==
Problem Number: 286 URL: https://leetcode.com/problems/count-good-triplets Title: 1534. Count Good Triplets Problem Description: Given an array of integers arr, and three integers a, b and c. You need to find the number of good triplets. A triplet (arr[i], arr[j], arr[k]) is good if the following conditions are true:

0 <= i < j < k < arr.length |arr[i] - arr[j]| <= a |arr[j] - arr[k]| <= b |arr[i] - arr[k]| <= c

Where |x| denotes the absolute value of x. Return the number of good triplets. Example 1: Input: arr = [3,0,1,1,9,7], a = 7, b = 2, c = 3 Output: 4 Explanation: There are 4 good triplets: [(3,0,1), (3,0,1), (3,1,1), (0,1,1)].

Example 2: Input: arr = [1,1,2,2,3], a = 0, b = 0, c = 1 Output: 0 Explanation: No triplet satisfies all conditions.

Constraints:

3 <= arr.length <= 100 0 <= arr[i] <= 1000 0 <= a, b, c <= 1000

==

Problem Number: 287 URL: https://leetcode.com/problems/kth-missing-positive-number Title: 1539. Kth Missing Positive Number Problem Description: Given an array arr of positive integers sorted in a strictly increasing order, and an integer k. Return the kth positive integer that is missing from this array. Example 1: Input: arr = [2,3,4,7,11], k = 5 Output: 9 Explanation: The missing positive integers are [1,5,6,8,9,10,12,13,...]. The 5th missing positive integer is 9.

Example 2: Input: arr = [1,2,3,4], k = 2 Output: 6 Explanation: The missing positive integers are [5,6,7,...]. The 2nd missing positive integer is 6.

Constraints:

1 <= arr.length <= 1000 1 <= arr[i] <= 1000 1 <= k <= 1000 arr[i] < arr[j] for 1 <= i < j <= arr.length

Follow up: Could you solve this problem in less than O(n) complexity?

==

Problem Number: 288 URL: https://leetcode.com/problems/make-the-string-great Title: 1544. Make The String Great Problem Description: Given a string s of lower and upper case English letters. A good string is a string which doesn't have two adjacent characters s[i] and s[i + 1] where:

0 <= i <= s.length - 2 s[i] is a lower-case letter and s[i + 1] is the same letter but in upper-case or vice-versa.

To make the string good, you can choose two adjacent characters that make the string bad and remove them. You can keep doing this until the string becomes good. Return the string after making it good. The answer is guaranteed to be unique under the given constraints. Notice that an empty string is also good. Example 1: Input: s = "leEeetcode" Output: "leetcode" Explanation: In the first step, either you choose i = 1 or i = 2, both will result "leEeetcode" to be reduced to "leetcode".

Example 2: Input: s = "abBAcC" Output: "" Explanation: We have many possible scenarios, and all lead to the same answer. For example: "abBAcC" --> "aAcC" --> "cC" --> "" "abBAcC" --> "abBA" --> "aA" --> ""

Example 3: Input: s = "s" Output: "s"

Constraints:

1 <= s.length <= 100 s contains only lower and upper case English letters.

==

Problem Number: 289 URL: https://leetcode.com/problems/three-consecutive-odds Title: 1550. Three Consecutive Odds Problem Description: Given an integer array arr, return true if there are three consecutive odd numbers in the array. Otherwise, return false. Example 1: Input: arr = [2,6,4,1] Output: false Explanation: There are no three consecutive odds.

Example 2: Input: arr = [1,2,34,3,4,5,7,23,12] Output: true Explanation: [5,7,23] are three consecutive odds.

Constraints:

1 <= arr.length <= 1000 1 <= arr[i] <= 1000

==

Problem Number: 290 URL: https://leetcode.com/problems/thousand-separator Title: 1556. Thousand Separator Problem Description: Given an integer n, add a dot (".") as the thousands separator and return it in string format. Example 1: Input: n = 987 Output: "987"

Example 2: Input: n = 1234 Output: "1.234"

Constraints:

0 <= n <= 231 - 1

==

Problem Number: 291 URL: https://leetcode.com/problems/most-visited-sector-in-a-circular-track Title: 1560. Most Visited Sector in a Circular Track Problem Description: Given an integer n and an integer array rounds. We have a circular track which consists of n sectors labeled from 1 to n. A marathon will be held on this track, the marathon consists of m rounds. The ith round starts at sector rounds[i - 1] and ends at sector rounds[i]. For example, round 1 starts at sector rounds[0] and ends at sector rounds[1] Return an array of the most visited sectors sorted in ascending order. Notice that you circulate the track in ascending order of sector numbers in the counter-clockwise direction (See the first example). Example 1:

Input: n = 4, rounds = [1,3,1,2] Output: [1,2] Explanation: The marathon starts at sector 1. The order of the visited sectors is as follows: 1 --> 2 --> 3 (end of round 1) --> 4 --> 1 (end of round 2) --> 2 (end of round 3 and the marathon) We can see that both sectors 1 and 2 are visited twice and they are the most visited sectors. Sectors 3 and 4 are visited only once. Example 2: Input: n = 2, rounds = [2,1,2,1,2,1,2,1,2] Output: [2]

Example 3: Input: n = 7, rounds = [1,3,5,7] Output: [1,2,3,4,5,6,7]

Constraints:

2 <= n <= 100 1 <= m <= 100 rounds.length == m + 1 1 <= rounds[i] <= n rounds[i] != rounds[i + 1] for 0 <= i < m

==
Problem Number: 292 URL: https://leetcode.com/problems/detect-pattern-of-length-m-repeated-k-or-more-times Title: 1566. Detect Pattern of Length M Repeated K or More Times Problem Description: Given an array of positive integers arr, find a pattern of length m that is repeated k or more times. A pattern is a subarray (consecutive sub-sequence) that consists of one or more values, repeated multiple times consecutively without overlapping. A pattern is defined by its length and the number of repetitions. Return true if there exists a pattern of length m that is repeated k or more times, otherwise return false. Example 1: Input: arr = [1,2,4,4,4,4], m = 1, k = 3 Output: true Explanation: The pattern (4) of length 1 is repeated 4 consecutive times. Notice that pattern can be repeated k or more times but not less.

Example 2: Input: arr = [1,2,1,2,1,1,1,3], m = 2, k = 2 Output: true Explanation: The pattern (1,2) of length 2 is repeated 2 consecutive times. Another valid pattern (2,1) is also repeated 2 times.

Example 3: Input: arr = [1,2,1,2,1,3], m = 2, k = 3 Output: false Explanation: The pattern (1,2) is of length 2 but is repeated only 2 times. There is no pattern of length 2 that is repeated 3 or more times.

Constraints:

2 <= arr.length <= 100 1 <= arr[i] <= 100 1 <= m <= 100 2 <= k <= 100
==
Problem Number: 293 URL: https://leetcode.com/problems/matrix-diagonal-sum Title: 1572. Matrix Diagonal Sum Problem Description: Given a square matrix mat, return the sum of the matrix diagonals. Only include the sum of all the elements on the primary diagonal and all the elements on the secondary diagonal that are not part of the primary diagonal. Example 1:

Input: mat = [[1,2,3], [4,5,6], [7,8,9]] Output: 25 Explanation: Diagonals sum: 1 + 5 + 9 + 3 + 7 = 25 Notice that element mat[1][1] = 5 is counted only once.

Example 2: Input: mat = [[1,1,1,1], [1,1,1,1], [1,1,1,1], [1,1,1,1]] Output: 8

Example 3: Input: mat = [[5]] Output: 5

Constraints:

n == mat.length == mat[i].length 1 <= n <= 100 1 <= mat[i][j] <= 100
==
Problem Number: 294 URL: https://leetcode.com/problems/replace-all-s-to-avoid-consecutive-repeating-characters Title: 1576. Replace All ?'s to Avoid Consecutive Repeating Characters Problem Description: Given a string s containing only lowercase English letters and the '?' character, convert all the '?' characters into lowercase letters such that the final string does not contain any consecutive repeating characters. You cannot modify the non '?'

characters. It is guaranteed that there are no consecutive repeating characters in the given string except for '?'. Return the final string after all the conversions (possibly zero) have been made. If there is more than one solution, return any of them. It can be shown that an answer is always possible with the given constraints. Example 1: Input: s = "?zs" Output: "azs" Explanation: There are 25 solutions for this problem. From "azs" to "yzs", all are valid. Only "z" is an invalid modification as the string will consist of consecutive repeating characters in "zzs".

Example 2: Input: s = "ubv?w" Output: "ubvaw" Explanation: There are 24 solutions for this problem. Only "v" and "w" are invalid modifications as the strings will consist of consecutive repeating characters in "ubvvw" and "ubvww".

Constraints:

1 <= s.length <= 100 s consist of lowercase English letters and '?'.

==
Problem Number: 295 URL: https://leetcode.com/problems/special-positions-in-a-binary-matrix Title: 1582. Special Positions in a Binary Matrix Problem Description: Given an m x n binary matrix mat, return the number of special positions in mat. A position (i, j) is called special if mat[i][j] == 1 and all other elements in row i and column j are 0 (rows and columns are 0-indexed). Example 1:

Input: mat = [[1,0,0],[0,0,1],[1,0,0]] Output: 1 Explanation: (1, 2) is a special position because mat[1][2] == 1 and all other elements in row 1 and column 2 are 0.

Example 2:

Input: mat = [[1,0,0],[0,1,0],[0,0,1]] Output: 3 Explanation: (0, 0), (1, 1) and (2, 2) are special positions.

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 100 mat[i][j] is either 0 or 1.

==
Problem Number: 296 URL: https://leetcode.com/problems/sum-of-all-odd-length-subarrays Title: 1588. Sum of All Odd Length Subarrays Problem Description: Given an array of positive integers arr, return the sum of all possible odd-length subarrays of arr. A subarray is a contiguous subsequence of the array. Example 1: Input: arr = [1,4,2,5,3] Output: 58 Explanation: The odd-length subarrays of arr and their sums are: [1] = 1 [4] = 4 [2] = 2 [5] = 5 [3] = 3 [1,4,2] = 7 [4,2,5] = 11 [2,5,3] = 10 [1,4,2,5,3] = 15 If we add all these together we get 1 + 4 + 2 + 5 + 3 + 7 + 11 + 10 + 15 = 58 Example 2: Input: arr = [1,2] Output: 3 Explanation: There are only 2 subarrays of odd length, [1] and [2]. Their sum is 3. Example 3: Input: arr = [10,11,12] Output: 66

Constraints:

1 <= arr.length <= 100 1 <= arr[i] <= 1000

Follow up: Could you solve this problem in O(n) time complexity?

==
Problem Number: 297 URL: https://leetcode.com/problems/rearrange-spaces-between-words Title: 1592. Rearrange Spaces Between Words Problem Description: You are given a string text of words that are placed among some number of spaces. Each word consists of one or more lowercase English letters and are separated by at least one space. It's guaranteed that text contains at least one word. Rearrange the spaces so that there is an equal number of spaces between every pair of adjacent words and that number is maximized. If you cannot redistribute all the spaces equally, place the extra spaces at the end, meaning the returned string should be the same length as text. Return the string after rearranging the spaces. Example 1: Input: text = " this is a sentence " Output: "this is a sentence" Explanation: There are a total of 9 spaces and 4 words. We can evenly divide the 9 spaces between the words: 9 / (4-1) = 3 spaces.

Example 2: Input: text = " practice makes perfect" Output: "practice makes perfect " Explanation: There are a total of 7 spaces and 3 words. 7 / (3-1) = 3 spaces plus 1 extra space. We place this extra space at the end of the string.

Constraints:

1 <= text.length <= 100 text consists of lowercase English letters and ' '. text contains at least one word.

==
Problem Number: 298 URL: https://leetcode.com/problems/crawler-log-folder Title: 1598. Crawler Log Folder Problem Description: The Leetcode file system keeps a log each time some user performs a change folder operation. The operations are described below:

"../" : Move to the parent folder of the current folder. (If you are already in the main folder, remain in the same folder). "./" : Remain in the same folder. "x/" : Move to the child folder named x (This folder is guaranteed to always exist).

You are given a list of strings logs where logs[i] is the operation performed by the user at the ith step. The file system starts in the main folder, then the operations in logs are performed. Return the minimum number of operations needed to go back to the main folder after the change folder operations. Example 1:

Input: logs = ["d1/","d2/","../","d21/","./"] Output: 2 Explanation: Use this change folder operation "../" 2 times and go back to the main folder.

Example 2:

Input: logs = ["d1/","d2/","./","d3/","../","d31/"] Output: 3

Example 3: Input: logs = ["d1/","../","../","../"] Output: 0

Constraints:

1 <= logs.length <= 103 2 <= logs[i].length <= 10 logs[i] contains lowercase English letters, digits, '.', and '/'. logs[i] follows the format described in the statement. Folder names consist of lowercase English letters and digits.

==

Problem Number: 299 URL: https://leetcode.com/problems/design-parking-system Title: 1603. Design Parking System Problem Description: Design a parking system for a parking lot. The parking lot has three kinds of parking spaces: big, medium, and small, with a fixed number of slots for each size. Implement the ParkingSystem class:

ParkingSystem(int big, int medium, int small) Initializes object of the ParkingSystem class. The number of slots for each parking space are given as part of the constructor. bool addCar(int carType) Checks whether there is a parking space of carType for the car that wants to get into the parking lot. carType can be of three kinds: big, medium, or small, which are represented by 1, 2, and 3 respectively. A car can only park in a parking space of its carType. If there is no space available, return false, else park the car in that size space and return true.

Example 1: Input ["ParkingSystem", "addCar", "addCar", "addCar", "addCar"] [[1, 1, 0], [1], [2], [3], [1]] Output [null, true, true, false, false]

Explanation ParkingSystem parkingSystem = new ParkingSystem(1, 1, 0); parkingSystem.addCar(1); // return true because there is 1 available slot for a big car parkingSystem.addCar(2); // return true because there is 1 available slot for a medium car parkingSystem.addCar(3); // return false because there is no available slot for a small car parkingSystem.addCar(1); // return false because there is no available slot for a big car. It is already occupied.

Constraints:

0 <= big, medium, small <= 1000 carType is 1, 2, or 3 At most 1000 calls will be made to addCar

==

Problem Number: 300 URL: https://leetcode.com/problems/special-array-with-x-elements-greater-than-or-equal-x Title: 1608. Special Array With X Elements Greater Than or Equal X Problem Description: You are given an array nums of non-negative integers. nums is considered special if there exists a number x such that there are exactly x numbers in nums that are greater than or equal to x. Notice that x does not have to be an element in nums. Return x if the array is special, otherwise, return -1. It can be proven that if nums is special, the value for x is unique. Example 1: Input: nums = [3,5] Output: 2 Explanation: There are 2 values (3 and 5) that are greater than or equal to 2.

Example 2: Input: nums = [0,0] Output: -1 Explanation: No numbers fit the criteria for x. If x = 0, there should be 0 numbers >= x, but there are 2. If x = 1, there should be 1 number >= x, but there are 0. If x = 2, there should be 2 numbers >= x, but there are 0. x cannot be greater since there are only 2 numbers in nums.

Example 3: Input: nums = [0,4,3,0,4] Output: 3 Explanation: There are 3 values that are greater than or equal to 3.

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 1000

==
Problem Number: 301 URL: https://leetcode.com/problems/maximum-nesting-depth-of-the-parentheses Title: 1614. Maximum Nesting Depth of the Parentheses Problem Description: A string is a valid parentheses string (denoted VPS) if it meets one of the following:

It is an empty string "", or a single character not equal to "(" or ")", It can be written as AB (A concatenated with B), where A and B are VPS's, or It can be written as (A), where A is a VPS.

We can similarly define the nesting depth depth(S) of any VPS S as follows:

depth("") = 0 depth(C) = 0, where C is a string with a single character not equal to "(" or ")". depth(A + B) = max(depth(A), depth(B)), where A and B are VPS's. depth("(" + A + ")") = 1 + depth(A), where A is a VPS.

For example, "", "()()", and "()(()())" are VPS's (with nesting depths 0, 1, and 2), and ")(" and "(()" are not VPS's. Given a VPS represented as string s, return the nesting depth of s. Example 1: Input: s = "(1+(2*3)+((8)/4))+1" Output: 3 Explanation: Digit 8 is inside of 3 nested parentheses in the string.

Example 2: Input: s = "(1)+((2))+(((3)))" Output: 3

Constraints:

1 <= s.length <= 100 s consists of digits 0-9 and characters '+', '-', '*', '/', '(', and ')'. It is guaranteed that parentheses expression s is a VPS.

==
Problem Number: 302 URL: https://leetcode.com/problems/mean-of-array-after-removing-some-elements Title: 1619. Mean of Array After Removing Some Elements Problem Description: Given an integer array arr, return the mean of the remaining integers after removing the smallest 5% and the largest 5% of the elements. Answers within 10-5 of the actual answer will be considered accepted. Example 1: Input: arr = [1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3] Output: 2.00000 Explanation: After erasing the minimum and the maximum values of this array, all elements are equal to 2, so the mean is 2.

Example 2: Input: arr = [6,2,7,5,1,2,0,3,10,2,5,0,5,5,0,8,7,6,8,0] Output: 4.00000

Example 3: Input: arr = [6,0,7,0,7,5,7,8,3,4,0,7,8,1,6,8,1,1,2,4,8,1,9,5,4,3,8,5,10,8,6,6,1,0,6,10,8,2,3,4] Output: 4.77778

Constraints:

20 <= arr.length <= 1000 arr.length is a multiple of 20. 0 <= arr[i] <= 105

==
Problem Number: 303 URL: https://leetcode.com/problems/largest-substring-between-two-equal-characters Title: 1624. Largest Substring Between Two Equal Characters Problem Description: Given a string s, return the length of the longest substring between two equal characters, excluding the two characters. If there is no such substring return -1. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "aa" Output: 0 Explanation: The optimal substring here is an empty substring between the two 'a's. Example 2: Input: s = "abca" Output: 2 Explanation: The optimal substring here is "bc".

Example 3: Input: s = "cbzxy" Output: -1 Explanation: There are no characters that appear twice in s.

Constraints:

1 <= s.length <= 300 s contains only lowercase English letters.

==
Problem Number: 304 URL: https://leetcode.com/problems/slowest-key Title: 1629. Slowest Key Problem Description: A newly designed keypad was tested, where a tester pressed a sequence of n keys, one at a time. You are given a string keysPressed of length n, where keysPressed[i] was the ith key pressed in the testing sequence, and a sorted list releaseTimes, where releaseTimes[i] was the time the ith key was released. Both arrays are 0-indexed. The 0th key was pressed at the time 0, and every subsequent key was pressed at the exact time the previous key was released. The tester wants to know the key of the keypress that had the longest duration. The ith keypress had a duration of releaseTimes[i] - releaseTimes[i - 1], and the 0th keypress had a duration of releaseTimes[0]. Note that the same key could have been pressed multiple times during the test, and these multiple presses of the same key may not have had the same duration. Return the key of the keypress that had the longest duration. If there are multiple such keypresses, return the lexicographically largest key of the keypresses. Example 1: Input: releaseTimes = [9,29,49,50], keysPressed = "cbcd" Output: "c" Explanation: The keypresses were as follows: Keypress for 'c' had a duration of 9 (pressed at time 0 and released at time 9). Keypress for 'b' had a duration of 29 - 9 = 20 (pressed at time 9 right after the release of the previous character and released at time 29). Keypress for 'c' had a duration of 49 - 29 = 20 (pressed at time 29 right after the release of the previous character and released at time 49). Keypress for

'd' had a duration of 50 - 49 = 1 (pressed at time 49 right after the release of the previous character and released at time 50). The longest of these was the keypress for 'b' and the second keypress for 'c', both with duration 20. 'c' is lexicographically larger than 'b', so the answer is 'c'.

Example 2: Input: releaseTimes = [12,23,36,46,62], keysPressed = "spuda" Output: "a" Explanation: The keypresses were as follows: Keypress for 's' had a duration of 12. Keypress for 'p' had a duration of 23 - 12 = 11. Keypress for 'u' had a duration of 36 - 23 = 13. Keypress for 'd' had a duration of 46 - 36 = 10. Keypress for 'a' had a duration of 62 - 46 = 16. The longest of these was the keypress for 'a' with duration 16. Constraints:

releaseTimes.length == n keysPressed.length == n 2 <= n <= 1000 1 <= releaseTimes[i] <= 109 releaseTimes[i] < releaseTimes[i+1] keysPressed contains only lowercase English letters.

==
Problem Number: 305 URL: https://leetcode.com/problems/sort-array-by-increasing-frequency Title: 1636. Sort Array by Increasing Frequency Problem Description: Given an array of integers nums, sort the array in increasing order based on the frequency of the values. If multiple values have the same frequency, sort them in decreasing order. Return the sorted array. Example 1: Input: nums = [1,1,2,2,2,3] Output: [3,1,1,2,2,2] Explanation: '3' has a frequency of 1, '1' has a frequency of 2, and '2' has a frequency of 3.

Example 2: Input: nums = [2,3,1,3,2] Output: [1,3,3,2,2] Explanation: '2' and '3' both have a frequency of 2, so they are sorted in decreasing order.

Example 3: Input: nums = [-1,1,-6,4,5,-6,1,4,1] Output: [5,-1,4,4,-6,-6,1,1,1] Constraints:

1 <= nums.length <= 100 -100 <= nums[i] <= 100

==
Problem Number: 306 URL: https://leetcode.com/problems/check-array-formation-through-concatenation Title: 1640. Check Array Formation Through Concatenation Problem Description: You are given an array of distinct integers arr and an array of integer arrays pieces, where the integers in pieces are distinct. Your goal is to form arr by concatenating the arrays in pieces in any order. However, you are not allowed to reorder the integers in each array pieces[i]. Return true if it is possible to form the array arr from pieces. Otherwise, return false. Example 1: Input: arr = [15,88], pieces = [[88],[15]] Output: true Explanation: Concatenate [15] then [88]

Example 2: Input: arr = [49,18,16], pieces = [[16,18,49]] Output: false Explanation: Even though the numbers match, we cannot reorder pieces[0].

Example 3: Input: arr = [91,4,64,78], pieces = [[78],[4,64],[91]] Output: true Explanation: Concatenate [91] then [4,64] then [78]

Constraints:

1 <= pieces.length <= arr.length <= 100 sum(pieces[i].length) == arr.length 1 <= pieces[i].length <= arr.length 1 <= arr[i], pieces[i][j] <= 100 The integers in arr are distinct. The integers in pieces are distinct (i.e., If we flatten pieces in a 1D array, all the integers in this array are distinct).

==

Problem Number: 307 URL: https://leetcode.com/problems/get-maximum-in-generated-array Title: 1646. Get Maximum in Generated Array Problem Description: You are given an integer n. A 0-indexed integer array nums of length n + 1 is generated in the following way:

nums[0] = 0 nums[1] = 1 nums[2 * i] = nums[i] when 2 <= 2 * i <= n nums[2 * i + 1] = nums[i] + nums[i + 1] when 2 <= 2 * i + 1 <= n

Return the maximum integer in the array nums. Example 1: Input: n = 7 Output: 3 Explanation: According to the given rules: nums[0] = 0 nums[1] = 1 nums[(1 * 2) = 2] = nums[1] = 1 nums[(1 * 2) + 1 = 3] = nums[1] + nums[2] = 1 + 1 = 2 nums[(2 * 2) = 4] = nums[2] = 1 nums[(2 * 2) + 1 = 5] = nums[2] + nums[3] = 1 + 2 = 3 nums[(3 * 2) = 6] = nums[3] = 2 nums[(3 * 2) + 1 = 7] = nums[3] + nums[4] = 2 + 1 = 3 Hence, nums = [0,1,1,2,1,3,2,3], and the maximum is max(0,1,1,2,1,3,2,3) = 3.

Example 2: Input: n = 2 Output: 1 Explanation: According to the given rules, nums = [0,1,1]. The maximum is max(0,1,1) = 1.

Example 3: Input: n = 3 Output: 2 Explanation: According to the given rules, nums = [0,1,1,2]. The maximum is max(0,1,1,2) = 2.

Constraints:

0 <= n <= 100

==

Problem Number: 308 URL: https://leetcode.com/problems/defuse-the-bomb Title: 1652. Defuse the Bomb Problem Description: You have a bomb to defuse, and your time is running out! Your informer will provide you with a circular array code of length of n and a key k. To decrypt the code, you must replace every number. All the numbers are replaced simultaneously.

If k > 0, replace the ith number with the sum of the next k numbers. If k < 0, replace the ith number with the sum of the previous k numbers. If k == 0, replace the ith number with 0.

As code is circular, the next element of code[n-1] is code[0], and the previous element of code[0] is code[n-1]. Given the circular array code and an integer key k, return the decrypted code to defuse the bomb! Example 1: Input: code = [5,7,1,4], k = 3 Output: [12,10,16,13] Explanation: Each number is replaced by the sum of the next 3 numbers. The decrypted code is [7+1+4, 1+4+5, 4+5+7, 5+7+1]. Notice that the numbers wrap around.

Example 2: Input: code = [1,2,3,4], k = 0 Output: [0,0,0,0] Explanation: When k is zero, the numbers are replaced by 0.

Example 3: Input: code = [2,4,9,3], k = -2 Output: [12,5,6,13] Explanation: The decrypted code is [3+9, 2+3, 4+2, 9+4]. Notice that the numbers wrap around again. If k is negative, the sum is of the previous numbers.

Constraints:

n == code.length 1 <= n <= 100 1 <= code[i] <= 100 -(n - 1) <= k <= n - 1

==
Problem Number: 309 URL: https://leetcode.com/problems/design-an-ordered-stream Title: 1656. Design an Ordered Stream Problem Description: There is a stream of n (idKey, value) pairs arriving in an arbitrary order, where idKey is an integer between 1 and n and value is a string. No two pairs have the same id. Design a stream that returns the values in increasing order of their IDs by returning a chunk (list) of values after each insertion. The concatenation of all the chunks should result in a list of the sorted values. Implement the OrderedStream class:

OrderedStream(int n) Constructs the stream to take n values. String[] insert(int idKey, String value) Inserts the pair (idKey, value) into the stream, then returns the largest possible chunk of currently inserted values that appear next in the order.

Example:

Input ["OrderedStream", "insert", "insert", "insert", "insert", "insert"] [[5], [3, "ccccc"], [1, "aaaaa"], [2, "bbbbb"], [5, "eeeee"], [4, "ddddd"]] Output [null, [], ["aaaaa"], ["bbbbb", "ccccc"], [], ["ddddd", "eeeee"]]

Explanation // Note that the values ordered by ID is ["aaaaa", "bbbbb", "ccccc", "ddddd", "eeeee"]. OrderedStream os = new OrderedStream(5); os.insert(3, "ccccc"); // Inserts (3, "ccccc"), returns []. os.insert(1, "aaaaa"); // Inserts (1, "aaaaa"), returns ["aaaaa"]. os.insert(2, "bbbbb"); // Inserts (2, "bbbbb"), returns ["bbbbb", "ccccc"]. os.insert(5, "eeeee"); // Inserts (5, "eeeee"), returns []. os.insert(4, "ddddd"); // Inserts (4, "ddddd"), returns ["ddddd", "eeeee"]. // Concatentating all the chunks returned: // [] + ["aaaaa"] + ["bbbbb", "ccccc"] + [] + ["ddddd", "eeeee"] = ["aaaaa", "bbbbb", "ccccc", "ddddd", "eeeee"] // The resulting order is the same as the order above.

Constraints:

1 <= n <= 1000 1 <= id <= n value.length == 5 value consists only of lowercase letters. Each call to insert will have a unique id. Exactly n calls will be made to insert.

==
Problem Number: 310 URL: https://leetcode.com/problems/check-if-two-string-arrays-are-equivalent Title: 1662. Check If Two String Arrays are

Equivalent Problem Description: Given two string arrays word1 and word2, return true if the two arrays represent the same string, and false otherwise. A string is represented by an array if the array elements concatenated in order forms the string. Example 1: Input: word1 = ["ab", "c"], word2 = ["a", "bc"] Output: true Explanation: word1 represents string "ab" + "c" -> "abc" word2 represents string "a" + "bc" -> "abc" The strings are the same, so return true. Example 2: Input: word1 = ["a", "cb"], word2 = ["ab", "c"] Output: false

Example 3: Input: word1 = ["abc", "d", "defg"], word2 = ["abcddefg"] Output: true

Constraints:

1 <= word1.length, word2.length <= 103 1 <= word1[i].length, word2[i].length <= 103 1 <= sum(word1[i].length), sum(word2[i].length) <= 103 word1[i] and word2[i] consist of lowercase letters.

==

Problem Number: 311 URL: https://leetcode.com/problems/maximum-repeating-substring Title: 1668. Maximum Repeating Substring Problem Description: For a string sequence, a string word is k-repeating if word concatenated k times is a substring of sequence. The word's maximum k-repeating value is the highest value k where word is k-repeating in sequence. If word is not a substring of sequence, word's maximum k-repeating value is 0. Given strings sequence and word, return the maximum k-repeating value of word in sequence. Example 1: Input: sequence = "ababc", word = "ab" Output: 2 Explanation: "abab" is a substring in "ababc".

Example 2: Input: sequence = "ababc", word = "ba" Output: 1 Explanation: "ba" is a substring in "ababc". "baba" is not a substring in "ababc".

Example 3: Input: sequence = "ababc", word = "ac" Output: 0 Explanation: "ac" is not a substring in "ababc".

Constraints:

1 <= sequence.length <= 100 1 <= word.length <= 100 sequence and word contains only lowercase English letters.

==

Problem Number: 312 URL: https://leetcode.com/problems/richest-customer-wealth Title: 1672. Richest Customer Wealth Problem Description: You are given an m x n integer grid accounts where accounts[i][j] is the amount of money the ith customer has in the jth bank. Return the wealth that the richest customer has. A customer's wealth is the amount of money they have in all their bank accounts. The richest customer is the customer that has the maximum wealth. Example 1: Input: accounts = [[1,2,3],[3,2,1]] Output: 6 Explanation: 1st customer has wealth = 1 + 2 + 3 = 6 2nd customer has wealth = 3 + 2 + 1 = 6 Both customers are considered the richest with a wealth of 6 each, so return 6.

Example 2: Input: accounts = [[1,5],[7,3],[3,5]] Output: 10 Explanation: 1st customer has wealth = 6 2nd customer has wealth = 10 3rd customer has wealth = 8 The 2nd customer is the richest with a wealth of 10. Example 3: Input: accounts = [[2,8,7],[7,1,3],[1,9,5]] Output: 17

Constraints:

m == accounts.length n == accounts[i].length 1 <= m, n <= 50 1 <= accounts[i][j] <= 100

==
Problem Number: 313 URL: https://leetcode.com/problems/goal-parser-interpretation Title: 1678. Goal Parser Interpretation Problem Description: You own a Goal Parser that can interpret a string command. The command consists of an alphabet of "G", "()" and/or "(al)" in some order. The Goal Parser will interpret "G" as the string "G", "()" as the string "o", and "(al)" as the string "al". The interpreted strings are then concatenated in the original order. Given the string command, return the Goal Parser's interpretation of command. Example 1: Input: command = "G()(al)" Output: "Goal" Explanation: The Goal Parser interprets the command as follows: G -> G () -> o (al) -> al The final concatenated result is "Goal".

Example 2: Input: command = "G()()()()(al)" Output: "Gooooal"

Example 3: Input: command = "(al)G(al)()()G" Output: "alGalooG"

Constraints:

1 <= command.length <= 100 command consists of "G", "()", and/or "(al)" in some order.

==
Problem Number: 314 URL: https://leetcode.com/problems/count-the-number-of-consistent-strings Title: 1684. Count the Number of Consistent Strings Problem Description: You are given a string allowed consisting of distinct characters and an array of strings words. A string is consistent if all characters in the string appear in the string allowed. Return the number of consistent strings in the array words. Example 1: Input: allowed = "ab", words = ["ad","bd","aaab","baa","badab"] Output: 2 Explanation: Strings "aaab" and "baa" are consistent since they only contain characters 'a' and 'b'.

Example 2: Input: allowed = "abc", words = ["a","b","c","ab","ac","bc","abc"] Output: 7 Explanation: All strings are consistent.

Example 3: Input: allowed = "cad", words = ["cc","acd","b","ba","bac","bad","ac","d"] Output: 4 Explanation: Strings "cc", "acd", "ac", and "d" are consistent.

Constraints:

1 <= words.length <= 104 1 <= allowed.length <= 26 1 <= words[i].length <= 10 The characters in allowed are distinct. words[i] and allowed contain only lowercase English letters.

==
Problem Number: 315 URL: https://leetcode.com/problems/count-of-matches-in-tournament Title: 1688. Count of Matches in Tournament Problem Description: You are given an integer n, the number of teams in a tournament that has strange rules:

If the current number of teams is even, each team gets paired with another team. A total of n / 2 matches are played, and n / 2 teams advance to the next round. If the current number of teams is odd, one team randomly advances in the tournament, and the rest gets paired. A total of (n - 1) / 2 matches are played, and (n - 1) / 2 + 1 teams advance to the next round.

Return the number of matches played in the tournament until a winner is decided. Example 1: Input: n = 7 Output: 6 Explanation: Details of the tournament: - 1st Round: Teams = 7, Matches = 3, and 4 teams advance. - 2nd Round: Teams = 4, Matches = 2, and 2 teams advance. - 3rd Round: Teams = 2, Matches = 1, and 1 team is declared the winner. Total number of matches = 3 + 2 + 1 = 6.

Example 2: Input: n = 14 Output: 13 Explanation: Details of the tournament: - 1st Round: Teams = 14, Matches = 7, and 7 teams advance. - 2nd Round: Teams = 7, Matches = 3, and 4 teams advance. - 3rd Round: Teams = 4, Matches = 2, and 2 teams advance. - 4th Round: Teams = 2, Matches = 1, and 1 team is declared the winner. Total number of matches = 7 + 3 + 2 + 1 = 13.

Constraints:

1 <= n <= 200

==
Problem Number: 316 URL: https://leetcode.com/problems/reformat-phone-number Title: 1694. Reformat Phone Number Problem Description: You are given a phone number as a string number. number consists of digits, spaces ' ', and/or dashes '-'. You would like to reformat the phone number in a certain manner. Firstly, remove all spaces and dashes. Then, group the digits from left to right into blocks of length 3 until there are 4 or fewer digits. The final digits are then grouped as follows:

2 digits: A single block of length 2. 3 digits: A single block of length 3. 4 digits: Two blocks of length 2 each.

The blocks are then joined by dashes. Notice that the reformatting process should never produce any blocks of length 1 and produce at most two blocks of length 2. Return the phone number after formatting. Example 1: Input: number = "1-23-45 6" Output: "123-456" Explanation: The digits are "123456". Step 1: There are more than 4 digits, so group the next 3 digits. The 1st block is "123". Step 2: There are 3 digits remaining, so put them in a single block of length 3. The 2nd block is "456". Joining the blocks gives "123-456".

Example 2: Input: number = "123 4-567" Output: "123-45-67" Explanation: The digits are "1234567". Step 1: There are more than 4 digits, so group the next 3 digits. The 1st block is "123". Step 2: There are 4 digits left, so split them into two blocks of length 2. The blocks are "45" and "67". Joining the blocks gives "123-45-67".

Example 3: Input: number = "123 4-5678" Output: "123-456-78" Explanation: The digits are "12345678". Step 1: The 1st block is "123". Step 2: The 2nd block is "456". Step 3: There are 2 digits left, so put them in a single block of length 2. The 3rd block is "78". Joining the blocks gives "123-456-78".

Constraints:

2 <= number.length <= 100 number consists of digits and the characters '-' and ' '. There are at least two digits in number.

==
Problem Number: 317 URL: https://leetcode.com/problems/number-of-students-unable-to-eat-lunch Title: 1700. Number of Students Unable to Eat Lunch Problem Description: The school cafeteria offers circular and square sandwiches at lunch break, referred to by numbers 0 and 1 respectively. All students stand in a queue. Each student either prefers square or circular sandwiches. The number of sandwiches in the cafeteria is equal to the number of students. The sandwiches are placed in a stack. At each step:

If the student at the front of the queue prefers the sandwich on the top of the stack, they will take it and leave the queue. Otherwise, they will leave it and go to the queue's end.

This continues until none of the queue students want to take the top sandwich and are thus unable to eat. You are given two integer arrays students and sandwiches where sandwiches[i] is the type of the ith sandwich in the stack (i = 0 is the top of the stack) and students[j] is the preference of the jth student in the initial queue (j = 0 is the front of the queue). Return the number of students that are unable to eat. Example 1: Input: students = [1,1,0,0], sandwiches = [0,1,0,1] Output: 0 Explanation: - Front student leaves the top sandwich and returns to the end of the line making students = [1,0,0,1]. - Front student leaves the top sandwich and returns to the end of the line making students = [0,0,1,1]. - Front student takes the top sandwich and leaves the line making students = [0,1,1] and sandwiches = [1,0,1]. - Front student leaves the top sandwich and returns to the end of the line making students = [1,1,0]. - Front student takes the top sandwich and leaves the line making students = [1,0] and sandwiches = [0,1]. - Front student leaves the top sandwich and returns to the end of the line making students = [0,1]. - Front student takes the top sandwich and leaves the line making students = [1] and sandwiches = [1]. - Front student takes the top sandwich and leaves the line making students = [] and sandwiches = []. Hence all students are able to eat.

Example 2: Input: students = [1,1,1,0,0,1], sandwiches = [1,0,0,0,1,1] Output:

3

Constraints:

1 <= students.length, sandwiches.length <= 100 students.length == sandwiches.length sandwiches[i] is 0 or 1. students[i] is 0 or 1.

==
Problem Number: 318 URL: https://leetcode.com/problems/determine-if-string-halves-are-alike Title: 1704. Determine if String Halves Are Alike Problem Description: You are given a string s of even length. Split this string into two halves of equal lengths, and let a be the first half and b be the second half. Two strings are alike if they have the same number of vowels ('a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U'). Notice that s contains uppercase and lowercase letters. Return true if a and b are alike. Otherwise, return false. Example 1: Input: s = "book" Output: true Explanation: a = "bo" and b = "ok". a has 1 vowel and b has 1 vowel. Therefore, they are alike.

Example 2: Input: s = "textbook" Output: false Explanation: a = "text" and b = "book". a has 1 vowel whereas b has 2. Therefore, they are not alike. Notice that the vowel o is counted twice.

Constraints:

2 <= s.length <= 1000 s.length is even. s consists of uppercase and lowercase letters.

==
Problem Number: 319 URL: https://leetcode.com/problems/maximum-units-on-a-truck Title: 1710. Maximum Units on a Truck Problem Description: You are assigned to put some amount of boxes onto one truck. You are given a 2D array boxTypes, where boxTypes[i] = [numberOfBoxesi, numberOfUnitsPerBoxi]:

numberOfBoxesi is the number of boxes of type i. numberOfUnitsPerBoxi is the number of units in each box of the type i.

You are also given an integer truckSize, which is the maximum number of boxes that can be put on the truck. You can choose any boxes to put on the truck as long as the number of boxes does not exceed truckSize. Return the maximum total number of units that can be put on the truck. Example 1: Input: boxTypes = [[1,3],[2,2],[3,1]], truckSize = 4 Output: 8 Explanation: There are: - 1 box of the first type that contains 3 units. - 2 boxes of the second type that contain 2 units each. - 3 boxes of the third type that contain 1 unit each. You can take all the boxes of the first and second types, and one box of the third type. The total number of units will be = (1 * 3) + (2 * 2) + (1 * 1) = 8.

Example 2: Input: boxTypes = [[5,10],[2,5],[4,7],[3,9]], truckSize = 10 Output: 91

Constraints:

1 <= boxTypes.length <= 1000 1 <= numberOfBoxesi, numberOfUnitsPerBoxi <= 1000 1 <= truckSize <= 106

==
Problem Number: 320 URL: https://leetcode.com/problems/calculate-money-in-leetcode-bank Title: 1716. Calculate Money in Leetcode Bank Problem Description: Hercy wants to save money for his first car. He puts money in the Leetcode bank every day. He starts by putting in $1 on Monday, the first day. Every day from Tuesday to Sunday, he will put in $1 more than the day before. On every subsequent Monday, he will put in $1 more than the previous Monday. Given n, return the total amount of money he will have in the Leetcode bank at the end of the nth day.   Example 1: Input: n = 4 Output: 10 Explanation: After the 4th day, the total is 1 + 2 + 3 + 4 = 10.

Example 2: Input: n = 10 Output: 37 Explanation: After the 10th day, the total is (1 + 2 + 3 + 4 + 5 + 6 + 7) + (2 + 3 + 4) = 37. Notice that on the 2nd Monday, Hercy only puts in $2.

Example 3: Input: n = 20 Output: 96 Explanation: After the 20th day, the total is (1 + 2 + 3 + 4 + 5 + 6 + 7) + (2 + 3 + 4 + 5 + 6 + 7 + 8) + (3 + 4 + 5 + 6 + 7 + 8) = 96.

Constraints:

1 <= n <= 1000

==
Problem Number: 321 URL: https://leetcode.com/problems/decode-xored-array Title: 1720. Decode XORed Array Problem Description: There is a hidden integer array arr that consists of n non-negative integers. It was encoded into another integer array encoded of length n - 1, such that encoded[i] = arr[i] XOR arr[i + 1]. For example, if arr = [1,0,2,1], then encoded = [1,2,3]. You are given the encoded array. You are also given an integer first, that is the first element of arr, i.e. arr[0]. Return the original array arr. It can be proved that the answer exists and is unique.   Example 1: Input: encoded = [1,2,3], first = 1 Output: [1,0,2,1] Explanation: If arr = [1,0,2,1], then first = 1 and encoded = [1 XOR 0, 0 XOR 2, 2 XOR 1] = [1,2,3]

Example 2: Input: encoded = [6,2,7,3], first = 4 Output: [4,2,0,7,4]

Constraints:

2 <= n <= 104 encoded.length == n - 1 0 <= encoded[i] <= 105 0 <= first <= 105

==
Problem Number: 322 URL: https://leetcode.com/problems/number-of-rectangles-that-can-form-the-largest-square Title: 1725. Number Of Rectangles That Can Form The Largest Square Problem Description: You are given an array rectangles where rectangles[i] = [li, wi] represents the ith rectangle of length li and width wi. You can cut the ith rectangle to form a square with

a side length of k if both k <= li and k <= wi. For example, if you have a rectangle [4,6], you can cut it to get a square with a side length of at most 4. Let maxLen be the side length of the largest square you can obtain from any of the given rectangles. Return the number of rectangles that can make a square with a side length of maxLen.   Example 1: Input: rectangles = [[5,8],[3,9],[5,12],[16,5]] Output: 3 Explanation: The largest squares you can get from each rectangle are of lengths [5,3,5,5]. The largest possible square is of length 5, and you can get it out of 3 rectangles.

Example 2: Input: rectangles = [[2,3],[3,7],[4,3],[3,7]] Output: 3

Constraints:

1 <= rectangles.length <= 1000 rectangles[i].length == 2 1 <= li, wi <= 109 li != wi

==

Problem Number: 323 URL: https://leetcode.com/problems/find-the-highest-altitude Title: 1732. Find the Highest Altitude Problem Description: There is a biker going on a road trip. The road trip consists of n + 1 points at different altitudes. The biker starts his trip on point 0 with altitude equal 0. You are given an integer array gain of length n where gain[i] is the net gain in altitude between points i and i + 1 for all (0 <= i < n). Return the highest altitude of a point.   Example 1: Input: gain = [-5,1,5,0,-7] Output: 1 Explanation: The altitudes are [0,-5,-4,1,1,-6]. The highest is 1.

Example 2: Input: gain = [-4,-3,-2,-1,4,3,2] Output: 0 Explanation: The altitudes are [0,-4,-7,-9,-10,-6,-3,-1]. The highest is 0.

Constraints:

n == gain.length 1 <= n <= 100 -100 <= gain[i] <= 100

==

Problem Number: 324 URL: https://leetcode.com/problems/latest-time-by-replacing-hidden-digits Title: 1736. Latest Time by Replacing Hidden Digits Problem Description: You are given a string time in the form of hh:mm, where some of the digits in the string are hidden (represented by ?). The valid times are those inclusively between 00:00 and 23:59. Return the latest valid time you can get from time by replacing the hidden digits.   Example 1: Input: time = "2?:?0" Output: "23:50" Explanation: The latest hour beginning with the digit '2' is 23 and the latest minute ending with the digit '0' is 50.

Example 2: Input: time = "0?:3?" Output: "09:39"

Example 3: Input: time = "1?:22" Output: "19:22"

Constraints:

time is in the format hh:mm. It is guaranteed that you can produce a valid time from the given string.

==
Problem Number: 325 URL: https://leetcode.com/problems/maximum-number-of-balls-in-a-box Title: 1742. Maximum Number of Balls in a Box Problem Description: You are working in a ball factory where you have n balls numbered from lowLimit up to highLimit inclusive (i.e., n == highLimit - lowLimit + 1), and an infinite number of boxes numbered from 1 to infinity. Your job at this factory is to put each ball in the box with a number equal to the sum of digits of the ball's number. For example, the ball number 321 will be put in the box number 3 + 2 + 1 = 6 and the ball number 10 will be put in the box number 1 + 0 = 1. Given two integers lowLimit and highLimit, return the number of balls in the box with the most balls. Example 1: Input: lowLimit = 1, highLimit = 10 Output: 2 Explanation: Box Number: 1 2 3 4 5 6 7 8 9 10 11 ... Ball Count: 2 1 1 1 1 1 1 1 1 0 0 ... Box 1 has the most number of balls with 2 balls. Example 2: Input: lowLimit = 5, highLimit = 15 Output: 2 Explanation: Box Number: 1 2 3 4 5 6 7 8 9 10 11 ... Ball Count: 1 1 1 1 2 2 1 1 1 0 0 ... Boxes 5 and 6 have the most number of balls with 2 balls in each.

Example 3: Input: lowLimit = 19, highLimit = 28 Output: 2 Explanation: Box Number: 1 2 3 4 5 6 7 8 9 10 11 12 ... Ball Count: 0 1 1 1 1 1 1 1 1 2 0 0 ... Box 10 has the most number of balls with 2 balls.

Constraints:

1 <= lowLimit <= highLimit <= 105

==
Problem Number: 326 URL: https://leetcode.com/problems/sum-of-unique-elements Title: 1748. Sum of Unique Elements Problem Description: You are given an integer array nums. The unique elements of an array are the elements that appear exactly once in the array. Return the sum of all the unique elements of nums. Example 1: Input: nums = [1,2,3,2] Output: 4 Explanation: The unique elements are [1,3], and the sum is 4.

Example 2: Input: nums = [1,1,1,1,1] Output: 0 Explanation: There are no unique elements, and the sum is 0.

Example 3: Input: nums = [1,2,3,4,5] Output: 15 Explanation: The unique elements are [1,2,3,4,5], and the sum is 15.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100

==
Problem Number: 327 URL: https://leetcode.com/problems/check-if-array-is-sorted-and-rotated Title: 1752. Check if Array Is Sorted and Rotated Problem Description: Given an array nums, return true if the array was originally sorted in non-decreasing order, then rotated some number of positions (including zero). Otherwise, return false. There may be duplicates in the original array. Note: An array A rotated by x positions results in an array B of the same length

such that A[i] == B[(i+x) % A.length], where % is the modulo operation. Example 1: Input: nums = [3,4,5,1,2] Output: true Explanation: [1,2,3,4,5] is the original sorted array. You can rotate the array by x = 3 positions to begin on the the element of value 3: [3,4,5,1,2].

Example 2: Input: nums = [2,1,3,4] Output: false Explanation: There is no sorted array once rotated that can make nums.

Example 3: Input: nums = [1,2,3] Output: true Explanation: [1,2,3] is the original sorted array. You can rotate the array by x = 0 positions (i.e. no rotation) to make nums.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100

==
Problem Number: 328 URL: https://leetcode.com/problems/minimum-changes-to-make-alternating-binary-string Title: 1758. Minimum Changes To Make Alternating Binary String Problem Description: You are given a string s consisting only of the characters '0' and '1'. In one operation, you can change any '0' to '1' or vice versa. The string is called alternating if no two adjacent characters are equal. For example, the string "010" is alternating, while the string "0100" is not. Return the minimum number of operations needed to make s alternating. Example 1: Input: s = "0100" Output: 1 Explanation: If you change the last character to '1', s will be "0101", which is alternating.

Example 2: Input: s = "10" Output: 0 Explanation: s is already alternating.

Example 3: Input: s = "1111" Output: 2 Explanation: You need two operations to reach "0101" or "1010".

Constraints:

1 <= s.length <= 104 s[i] is either '0' or '1'.

==
Problem Number: 329 URL: https://leetcode.com/problems/longest-nice-substring Title: 1763. Longest Nice Substring Problem Description: A string s is nice if, for every letter of the alphabet that s contains, it appears both in uppercase and lowercase. For example, "abABB" is nice because 'A' and 'a' appear, and 'B' and 'b' appear. However, "abA" is not because 'b' appears, but 'B' does not. Given a string s, return the longest substring of s that is nice. If there are multiple, return the substring of the earliest occurrence. If there are none, return an empty string. Example 1: Input: s = "YazaAay" Output: "aAa" Explanation: "aAa" is a nice string because 'A/a' is the only letter of the alphabet in s, and both 'A' and 'a' appear. "aAa" is the longest nice substring.

Example 2: Input: s = "Bb" Output: "Bb" Explanation: "Bb" is a nice string because both 'B' and 'b' appear. The whole string is a substring.

Example 3: Input: s = "c" Output: "" Explanation: There are no nice substrings.

Constraints:

1 <= s.length <= 100 s consists of uppercase and lowercase English letters.

==
Problem Number: 330 URL: https://leetcode.com/problems/merge-strings-alternately Title: 1768. Merge Strings Alternately Problem Description: You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string. Return the merged string. Example 1: Input: word1 = "abc", word2 = "pqr" Output: "apbqcr" Explanation: The merged string will be merged as so: word1: a b c word2: p q r merged: a p b q c r

Example 2: Input: word1 = "ab", word2 = "pqrs" Output: "apbqrs" Explanation: Notice that as word2 is longer, "rs" is appended to the end. word1: a b word2: p q r s merged: a p b q r s

Example 3: Input: word1 = "abcd", word2 = "pq" Output: "apbqcd" Explanation: Notice that as word1 is longer, "cd" is appended to the end. word1: a b c d word2: p q merged: a p b q c d

Constraints:

1 <= word1.length, word2.length <= 100 word1 and word2 consist of lowercase English letters.

==
Problem Number: 331 URL: https://leetcode.com/problems/count-items-matching-a-rule Title: 1773. Count Items Matching a Rule Problem Description: You are given an array items, where each items[i] = [typei, colori, namei] describes the type, color, and name of the ith item. You are also given a rule represented by two strings, ruleKey and ruleValue. The ith item is said to match the rule if one of the following is true:

ruleKey == "type" and ruleValue == typei. ruleKey == "color" and ruleValue == colori. ruleKey == "name" and ruleValue == namei.

Return the number of items that match the given rule. Example 1: Input: items = [["phone","blue","pixel"],["computer","silver","lenovo"],["phone","gold","iphone"]], ruleKey = "color", ruleValue = "silver" Output: 1 Explanation: There is only one item matching the given rule, which is ["computer","silver","lenovo"].

Example 2: Input: items = [["phone","blue","pixel"],["computer","silver","phone"],["phone","gold","iphone"]], ruleKey = "type", ruleValue = "phone" Output: 2 Explanation: There are only two items matching the given rule, which are ["phone","blue","pixel"] and ["phone","gold","iphone"]. Note that the item ["computer","silver","phone"] does not match. Constraints:

1 <= items.length <= 104 1 <= typei.length, colori.length, namei.length, ruleValue.length <= 10 ruleKey is equal to either "type", "color", or "name". All strings consist only of lowercase letters.

==
Problem Number: 332 URL: https://leetcode.com/problems/find-nearest-point-that-has-the-same-x-or-y-coordinate Title: 1779. Find Nearest Point That Has the Same X or Y Coordinate Problem Description: You are given two integers, x and y, which represent your current location on a Cartesian grid: (x, y). You are also given an array points where each points[i] = [ai, bi] represents that a point exists at (ai, bi). A point is valid if it shares the same x-coordinate or the same y-coordinate as your location. Return the index (0-indexed) of the valid point with the smallest Manhattan distance from your current location. If there are multiple, return the valid point with the smallest index. If there are no valid points, return -1. The Manhattan distance between two points (x1, y1) and (x2, y2) is abs(x1 - x2) + abs(y1 - y2). Example 1: Input: x = 3, y = 4, points = [[1,2],[3,1],[2,4],[2,3],[4,4]] Output: 2 Explanation: Of all the points, only [3,1], [2,4] and [4,4] are valid. Of the valid points, [2,4] and [4,4] have the smallest Manhattan distance from your current location, with a distance of 1. [2,4] has the smallest index, so return 2. Example 2: Input: x = 3, y = 4, points = [[3,4]] Output: 0 Explanation: The answer is allowed to be on the same location as your current location. Example 3: Input: x = 3, y = 4, points = [[2,3]] Output: -1 Explanation: There are no valid points. Constraints:

1 <= points.length <= 104 points[i].length == 2 1 <= x, y, ai, bi <= 104

==
Problem Number: 333 URL: https://leetcode.com/problems/check-if-binary-string-has-at-most-one-segment-of-ones Title: 1784. Check if Binary String Has at Most One Segment of Ones Problem Description: Given a binary string s without leading zeros, return true if s contains at most one contiguous segment of ones. Otherwise, return false. Example 1: Input: s = "1001" Output: false Explanation: The ones do not form a contiguous segment.

Example 2: Input: s = "110" Output: true Constraints:

1 <= s.length <= 100 s[i] is either '0' or '1'. s[0] is '1'.

==
Problem Number: 334 URL: https://leetcode.com/problems/check-if-one-string-swap-can-make-strings-equal Title: 1790. Check if One String Swap Can Make Strings Equal Problem Description: You are given two strings s1 and s2 of equal length. A string swap is an operation where you choose two indices in a string (not necessarily different) and swap the characters at these indices. Return true if it is possible to make both strings equal by performing at most one string swap on exactly one of the strings. Otherwise, return false. Example 1: Input: s1 = "bank", s2 = "kanb" Output: true Explanation: For example, swap the first character with the last character of s2 to make "bank".

Example 2: Input: s1 = "attack", s2 = "defend" Output: false Explanation: It is impossible to make them equal with one string swap.

Example 3: Input: s1 = "kelb", s2 = "kelb" Output: true Explanation: The two strings are already equal, so no string swap operation is required.

Constraints:

1 <= s1.length, s2.length <= 100 s1.length == s2.length s1 and s2 consist of only lowercase English letters.

==

Problem Number: 335 URL: https://leetcode.com/problems/find-center-of-star-graph Title: 1791. Find Center of Star Graph Problem Description: There is an undirected star graph consisting of n nodes labeled from 1 to n. A star graph is a graph where there is one center node and exactly n - 1 edges that connect the center node with every other node. You are given a 2D integer array edges where each edges[i] = [ui, vi] indicates that there is an edge between the nodes ui and vi. Return the center of the given star graph. Example 1:

Input: edges = [[1,2],[2,3],[4,2]] Output: 2 Explanation: As shown in the figure above, node 2 is connected to every other node, so 2 is the center.

Example 2: Input: edges = [[1,2],[5,1],[1,3],[1,4]] Output: 1

Constraints:

3 <= n <= 105 edges.length == n - 1 edges[i].length == 2 1 <= ui, vi <= n ui != vi The given edges represent a valid star graph.

==

Problem Number: 336 URL: https://leetcode.com/problems/second-largest-digit-in-a-string Title: 1796. Second Largest Digit in a String Problem Description: Given an alphanumeric string s, return the second largest numerical digit that appears in s, or -1 if it does not exist. An alphanumeric string is a string consisting of lowercase English letters and digits. Example 1: Input: s = "dfa12321afd" Output: 2 Explanation: The digits that appear in s are [1, 2, 3]. The second largest digit is 2.

Example 2: Input: s = "abc1111" Output: -1 Explanation: The digits that appear in s are [1]. There is no second largest digit.

Constraints:

1 <= s.length <= 500 s consists of only lowercase English letters and/or digits.

==

Problem Number: 337 URL: https://leetcode.com/problems/maximum-ascending-subarray-sum Title: 1800. Maximum Ascending Subarray Sum Problem Description: Given an array of positive integers nums, return the maximum possible sum of an ascending subarray in nums. A subarray is defined as a contiguous sequence of numbers in an array. A subarray [numsl,

numsl+1, ..., numsr-1, numsr] is ascending if for all i where l <= i < r, numsi < numsi+1. Note that a subarray of size 1 is ascending.   Example 1: Input: nums = [10,20,30,5,10,50] Output: 65 Explanation: [5,10,50] is the ascending subarray with the maximum sum of 65.

Example 2: Input: nums = [10,20,30,40,50] Output: 150 Explanation: [10,20,30,40,50] is the ascending subarray with the maximum sum of 150.

Example 3: Input: nums = [12,17,15,13,10,11,12] Output: 33 Explanation: [10,11,12] is the ascending subarray with the maximum sum of 33.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100

==

Problem Number: 338 URL: https://leetcode.com/problems/number-of-different-integers-in-a-string Title: 1805. Number of Different Integers in a String Problem Description: You are given a string word that consists of digits and lowercase English letters. You will replace every non-digit character with a space. For example, "a123bc34d8ef34" will become " 123 34 8 34". Notice that you are left with some integers that are separated by at least one space: "123", "34", "8", and "34". Return the number of different integers after performing the replacement operations on word. Two integers are considered different if their decimal representations without any leading zeros are different. Example 1: Input: word = "a123bc34d8ef34" Output: 3 Explanation: The three different integers are "123", "34", and "8". Notice that "34" is only counted once.

Example 2: Input: word = "leet1234code234" Output: 2

Example 3: Input: word = "a1b01c001" Output: 1 Explanation: The three integers "1", "01", and "001" all represent the same integer because the leading zeros are ignored when comparing their decimal values.

Constraints:

1 <= word.length <= 1000 word consists of digits and lowercase English letters.

==

Problem Number: 339 URL: https://leetcode.com/problems/determine-color-of-a-chessboard-square Title: 1812. Determine Color of a Chessboard Square Problem Description: You are given coordinates, a string that represents the coordinates of a square of the chessboard. Below is a chessboard for your reference.

Return true if the square is white, and false if the square is black. The coordinate will always represent a valid chessboard square. The coordinate will always have the letter first, and the number second.   Example 1: Input: coordinates = "a1" Output: false Explanation: From the chessboard above, the square with coordinates "a1" is black, so return false.

Example 2: Input: coordinates = ”h3” Output: true Explanation: From the chessboard above, the square with coordinates ”h3” is white, so return true.

Example 3: Input: coordinates = ”c7” Output: false

Constraints:

coordinates.length == 2 ’a’ <= coordinates[0] <= ’h’ ’1’ <= coordinates[1] <= ’8’

==
Problem Number: 340 URL: https://leetcode.com/problems/truncate-sentence Title: 1816. Truncate Sentence Problem Description: A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each of the words consists of only uppercase and lowercase English letters (no punctuation).

For example, ”Hello World”, ”HELLO”, and ”hello world hello world” are all sentences.

You are given a sentence s and an integer k. You want to truncate s such that it contains only the first k words. Return s after truncating it. Example 1: Input: s = ”Hello how are you Contestant”, k = 4 Output: ”Hello how are you” Explanation: The words in s are [”Hello”, ”how” ”are”, ”you”, ”Contestant”]. The first 4 words are [”Hello”, ”how”, ”are”, ”you”]. Hence, you should return ”Hello how are you”.

Example 2: Input: s = ”What is the solution to this problem”, k = 4 Output: ”What is the solution” Explanation: The words in s are [”What”, ”is” ”the”, ”solution”, ”to”, ”this”, ”problem”]. The first 4 words are [”What”, ”is”, ”the”, ”solution”]. Hence, you should return ”What is the solution”. Example 3: Input: s = ”chopper is not a tanuki”, k = 5 Output: ”chopper is not a tanuki”

Constraints:

1 <= s.length <= 500 k is in the range [1, the number of words in s]. s consist of only lowercase and uppercase English letters and spaces. The words in s are separated by a single space. There are no leading or trailing spaces.

==
Problem Number: 341 URL: https://leetcode.com/problems/sign-of-the-product-of-an-array Title: 1822. Sign of the Product of an Array Problem Description: There is a function signFunc(x) that returns:

1 if x is positive. -1 if x is negative. 0 if x is equal to 0.

You are given an integer array nums. Let product be the product of all values in the array nums. Return signFunc(product). Example 1: Input: nums = [-1,-2,-3,-4,3,2,1] Output: 1 Explanation: The product of all values in the array is 144, and signFunc(144) = 1

Example 2: Input: nums = [1,5,0,2,-3] Output: 0 Explanation: The product of all values in the array is 0, and signFunc(0) = 0

Example 3: Input: nums = [-1,1,-1,1,-1] Output: -1 Explanation: The product of all values in the array is -1, and signFunc(-1) = -1

Constraints:

1 <= nums.length <= 1000 -100 <= nums[i] <= 100

==
Problem Number: 342 URL: https://leetcode.com/problems/minimum-operations-to-make-the-array-increasing Title: 1827. Minimum Operations to Make the Array Increasing Problem Description: You are given an integer array nums (0-indexed). In one operation, you can choose an element of the array and increment it by 1.

For example, if nums = [1,2,3], you can choose to increment nums[1] to make nums = [1,3,3].

Return the minimum number of operations needed to make nums strictly increasing. An array nums is strictly increasing if nums[i] < nums[i+1] for all 0 <= i < nums.length - 1. An array of length 1 is trivially strictly increasing. Example 1: Input: nums = [1,1,1] Output: 3 Explanation: You can do the following operations: 1) Increment nums[2], so nums becomes [1,1,2]. 2) Increment nums[1], so nums becomes [1,2,2]. 3) Increment nums[2], so nums becomes [1,2,3].

Example 2: Input: nums = [1,5,2,4,1] Output: 14

Example 3: Input: nums = [8] Output: 0

Constraints:

1 <= nums.length <= 5000 1 <= nums[i] <= 104

==
Problem Number: 343 URL: https://leetcode.com/problems/check-if-the-sentence-is-pangram Title: 1832. Check if the Sentence Is Pangram Problem Description: A pangram is a sentence where every letter of the English alphabet appears at least once. Given a string sentence containing only lowercase English letters, return true if sentence is a pangram, or false otherwise. Example 1: Input: sentence = "thequickbrownfoxjumpsoverthelazydog" Output: true Explanation: sentence contains at least one of every letter of the English alphabet.

Example 2: Input: sentence = "leetcode" Output: false

Constraints:

1 <= sentence.length <= 1000 sentence consists of lowercase English letters.

==
Problem Number: 344 URL: https://leetcode.com/problems/sum-of-digits-in-base-k Title: 1837. Sum of Digits in Base K Problem Description: Given an integer n (in base 10) and a base k, return the sum of the digits of n after converting n from base 10 to base k. After converting, each digit should be interpreted as a base 10 number, and the sum should be returned in base 10. Example 1: Input: n = 34, k = 6 Output: 9 Explanation: 34 (base 10) expressed in base 6 is 54. 5 + 4 = 9.

Example 2: Input: n = 10, k = 10 Output: 1 Explanation: n is already in base 10. 1 + 0 = 1.

Constraints:

1 <= n <= 100 2 <= k <= 10
==
Problem Number: 345 URL: https://leetcode.com/problems/replace-all-digits-with-characters Title: 1844. Replace All Digits with Characters Problem Description: You are given a 0-indexed string s that has lowercase English letters in its even indices and digits in its odd indices. There is a function shift(c, x), where c is a character and x is a digit, that returns the xth character after c.

For example, shift('a', 5) = 'f' and shift('x', 0) = 'x'.

For every odd index i, you want to replace the digit s[i] with shift(s[i-1], s[i]). Return s after replacing all digits. It is guaranteed that shift(s[i-1], s[i]) will never exceed 'z'. Example 1: Input: s = "a1c1e1" Output: "abcdef" Explanation: The digits are replaced as follows: - s[1] -> shift('a',1) = 'b' - s[3] -> shift('c',1) = 'd' - s[5] -> shift('e',1) = 'f' Example 2: Input: s = "a1b2c3d4e" Output: "abbdcfdhe" Explanation: The digits are replaced as follows: - s[1] -> shift('a',1) = 'b' - s[3] -> shift('b',2) = 'd' - s[5] -> shift('c',3) = 'f' - s[7] -> shift('d',4) = 'h' Constraints:

1 <= s.length <= 100 s consists only of lowercase English letters and digits. shift(s[i-1], s[i]) <= 'z' for all odd indices i.
==
Problem Number: 346 URL: https://leetcode.com/problems/minimum-distance-to-the-target-element Title: 1848. Minimum Distance to the Target Element Problem Description: Given an integer array nums (0-indexed) and two integers target and start, find an index i such that nums[i] == target and abs(i - start) is minimized. Note that abs(x) is the absolute value of x. Return abs(i - start). It is guaranteed that target exists in nums. Example 1: Input: nums = [1,2,3,4,5], target = 5, start = 3 Output: 1 Explanation: nums[4] = 5 is the only value equal to target, so the answer is abs(4 - 3) = 1.

Example 2: Input: nums = [1], target = 1, start = 0 Output: 0 Explanation: nums[0] = 1 is the only value equal to target, so the answer is abs(0 - 0) = 0.

Example 3: Input: nums = [1,1,1,1,1,1,1,1,1,1], target = 1, start = 0 Output: 0 Explanation: Every value of nums is 1, but nums[0] minimizes abs(i - start), which is abs(0 - 0) = 0.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 104 0 <= start < nums.length target is in nums.

==

Problem Number: 347 URL: https://leetcode.com/problems/maximum-population-year Title: 1854. Maximum Population Year Problem Description: You are given a 2D integer array logs where each logs[i] = [birthi, deathi] indicates the birth and death years of the ith person. The population of some year x is the number of people alive during that year. The ith person is counted in year x's population if x is in the inclusive range [birthi, deathi - 1]. Note that the person is not counted in the year that they die. Return the earliest year with the maximum population. Example 1: Input: logs = [[1993,1999],[2000,2010]] Output: 1993 Explanation: The maximum population is 1, and 1993 is the earliest year with this population.

Example 2: Input: logs = [[1950,1961],[1960,1971],[1970,1981]] Output: 1960 Explanation: The maximum population is 2, and it had happened in years 1960 and 1970. The earlier year between them is 1960. Constraints:

1 <= logs.length <= 100 1950 <= birthi < deathi <= 2050

==

Problem Number: 348 URL: https://leetcode.com/problems/sorting-the-sentence Title: 1859. Sorting the Sentence Problem Description: A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters. A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence. Example 1: Input: s = "is2 sentence4 This1 a3" Output: "This is a sentence" Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2: Input: s = "Myself2 Me1 I4 and3" Output: "Me Myself and I" Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

2 <= s.length <= 200 s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9. The number of words in s is between 1 and 9.

The words in s are separated by a single space. s contains no leading or trailing spaces.

==
Problem Number: 349 URL: https://leetcode.com/problems/sum-of-all-subset-xor-totals Title: 1863. Sum of All Subset XOR Totals Problem Description: The XOR total of an array is defined as the bitwise XOR of all its elements, or 0 if the array is empty.

For example, the XOR total of the array [2,5,6] is 2 XOR 5 XOR 6 = 1.

Given an array nums, return the sum of all XOR totals for every subset of nums. Note: Subsets with the same elements should be counted multiple times. An array a is a subset of an array b if a can be obtained from b by deleting some (possibly zero) elements of b.   Example 1: Input: nums = [1,3] Output: 6 Explanation: The 4 subsets of [1,3] are: - The empty subset has an XOR total of 0. - [1] has an XOR total of 1. - [3] has an XOR total of 3. - [1,3] has an XOR total of 1 XOR 3 = 2. 0 + 1 + 3 + 2 = 6

Example 2: Input: nums = [5,1,6] Output: 28 Explanation: The 8 subsets of [5,1,6] are: - The empty subset has an XOR total of 0. - [5] has an XOR total of 5. - [1] has an XOR total of 1. - [6] has an XOR total of 6. - [5,1] has an XOR total of 5 XOR 1 = 4. - [5,6] has an XOR total of 5 XOR 6 = 3. - [1,6] has an XOR total of 1 XOR 6 = 7. - [5,1,6] has an XOR total of 5 XOR 1 XOR 6 = 2. 0 + 5 + 1 + 6 + 4 + 3 + 7 + 2 = 28

Example 3: Input: nums = [3,4,5,6,7,8] Output: 480 Explanation: The sum of all XOR totals for every subset is 480.

Constraints:

1 <= nums.length <= 12 1 <= nums[i] <= 20

==
Problem Number: 350 URL: https://leetcode.com/problems/longer-contiguous-segments-of-ones-than-zeros Title: 1869. Longer Contiguous Segments of Ones than Zeros Problem Description: Given a binary string s, return true if the longest contiguous segment of 1's is strictly longer than the longest contiguous segment of 0's in s, or return false otherwise.

For example, in s = "110100010" the longest continuous segment of 1s has length 2, and the longest continuous segment of 0s has length 3.

Note that if there are no 0's, then the longest continuous segment of 0's is considered to have a length 0. The same applies if there is no 1's.   Example 1: Input: s = "1101" Output: true Explanation: The longest contiguous segment of 1s has length 2: "1101" The longest contiguous segment of 0s has length 1: "1101" The segment of 1s is longer, so return true.

Example 2: Input: s = "111000" Output: false Explanation: The longest contiguous segment of 1s has length 3: "111000" The longest contiguous segment of 0s has length 3: "111000" The segment of 1s is not longer, so return false.

Example 3: Input: s = "110100010" Output: false Explanation: The longest contiguous segment of 1s has length 2: "110100010" The longest contiguous segment of 0s has length 3: "110100010" The segment of 1s is not longer, so return false.

Constraints:

1 <= s.length <= 100 s[i] is either '0' or '1'.

==

Problem Number: 351 URL: https://leetcode.com/problems/substrings-of-size-three-with-distinct-characters Title: 1876. Substrings of Size Three with Distinct Characters Problem Description: A string is good if there are no repeated characters. Given a string s, return the number of good substrings of length three in s. Note that if there are multiple occurrences of the same substring, every occurrence should be counted. A substring is a contiguous sequence of characters in a string. Example 1: Input: s = "xyzzaz" Output: 1 Explanation: There are 4 substrings of size 3: "xyz", "yzz", "zza", and "zaz". The only good substring of length 3 is "xyz".

Example 2: Input: s = "aababcabc" Output: 4 Explanation: There are 7 substrings of size 3: "aab", "aba", "bab", "abc", "bca", "cab", and "abc". The good substrings are "abc", "bca", "cab", and "abc".

Constraints:

1 <= s.length <= 100 s consists of lowercase English letters.

==

Problem Number: 352 URL: https://leetcode.com/problems/check-if-word-equals-summation-of-two-words Title: 1880. Check if Word Equals Summation of Two Words Problem Description: The letter value of a letter is its position in the alphabet starting from 0 (i.e. 'a' -> 0, 'b' -> 1, 'c' -> 2, etc.). The numerical value of some string of lowercase English letters s is the concatenation of the letter values of each letter in s, which is then converted into an integer.

For example, if s = "acb", we concatenate each letter's letter value, resulting in "021". After converting it, we get 21.

You are given three strings firstWord, secondWord, and targetWord, each consisting of lowercase English letters 'a' through 'j' inclusive. Return true if the summation of the numerical values of firstWord and secondWord equals the numerical value of targetWord, or false otherwise. Example 1: Input: firstWord = "acb", secondWord = "cba", targetWord = "cdb" Output: true Explanation: The numerical value of firstWord is "acb" -> "021" -> 21. The numerical value of secondWord is "cba" -> "210" -> 210. The numerical value of targetWord is "cdb" -> "231" -> 231. We return true because 21 + 210 == 231.

Example 2: Input: firstWord = "aaa", secondWord = "a", targetWord = "aab"

Output: false Explanation: The numerical value of firstWord is ”aaa” -> ”000” -> 0. The numerical value of secondWord is ”a” -> ”0” -> 0. The numerical value of targetWord is ”aab” -> ”001” -> 1. We return false because 0 + 0 != 1.

Example 3: Input: firstWord = ”aaa”, secondWord = ”a”, targetWord = ”aaaa” Output: true Explanation: The numerical value of firstWord is ”aaa” -> ”000” -> 0. The numerical value of secondWord is ”a” -> ”0” -> 0. The numerical value of targetWord is ”aaaa” -> ”0000” -> 0. We return true because 0 + 0 == 0.

Constraints:

1 <= firstWord.length, secondWord.length, targetWord.length <= 8 firstWord, secondWord, and targetWord consist of lowercase English letters from ’a’ to ’j’ inclusive.

==

Problem Number: 353 URL: https://leetcode.com/problems/determine-whether-matrix-can-be-obtained-by-rotation Title: 1886. Determine Whether Matrix Can Be Obtained By Rotation Problem Description: Given two n x n binary matrices mat and target, return true if it is possible to make mat equal to target by rotating mat in 90-degree increments, or false otherwise. Example 1:

Input: mat = [[0,1],[1,0]], target = [[1,0],[0,1]] Output: true Explanation: We can rotate mat 90 degrees clockwise to make mat equal target.

Example 2:

Input: mat = [[0,1],[1,1]], target = [[1,0],[0,1]] Output: false Explanation: It is impossible to make mat equal to target by rotating mat.

Example 3:

Input: mat = [[0,0,0],[0,1,0],[1,1,1]], target = [[1,1,1],[0,1,0],[0,0,0]] Output: true Explanation: We can rotate mat 90 degrees clockwise two times to make mat equal target.

Constraints:

n == mat.length == target.length n == mat[i].length == target[i].length 1 <= n <= 10 mat[i][j] and target[i][j] are either 0 or 1.

==

Problem Number: 354 URL: https://leetcode.com/problems/check-if-all-the-integers-in-a-range-are-covered Title: 1893. Check if All the Integers in a Range Are Covered Problem Description: You are given a 2D integer array ranges and two integers left and right. Each ranges[i] = [starti, endi] represents an inclusive interval between starti and endi. Return true if each integer in the inclusive range [left, right] is covered by at least one interval in ranges. Return false otherwise. An integer x is covered by an interval ranges[i] = [starti, endi] if starti <= x <= endi.   Example 1: Input: ranges = [[1,2],[3,4],[5,6]], left = 2, right = 5 Output: true Explanation: Every integer between 2 and 5 is covered: - 2 is covered by the first range. - 3 and 4 are covered by the second range. - 5 is covered by the third range.

Example 2: Input: ranges = [[1,10],[10,20]], left = 21, right = 21 Output: false Explanation: 21 is not covered by any range.

Constraints:

1 <= ranges.length <= 50 1 <= starti <= endi <= 50 1 <= left <= right <= 50

==

Problem Number: 355 URL: https://leetcode.com/problems/redistribute-characters-to-make-all-strings-equal Title: 1897. Redistribute Characters to Make All Strings Equal Problem Description: You are given an array of strings words (0-indexed). In one operation, pick two distinct indices i and j, where words[i] is a non-empty string, and move any character from words[i] to any position in words[j]. Return true if you can make every string in words equal using any number of operations, and false otherwise.   Example 1: Input: words = ["abc","aabc","bc"] Output: true Explanation: Move the first 'a' in words[1] to the front of words[2], to make words[1] = "abc" and words[2] = "abc". All the strings are now equal to "abc", so return true.

Example 2: Input: words = ["ab","a"] Output: false Explanation: It is impossible to make all the strings equal using the operation.

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 100 words[i] consists of lowercase English letters.

==

Problem Number: 356 URL: https://leetcode.com/problems/largest-odd-number-in-string Title: 1903. Largest Odd Number in String Problem Description: You are given a string num, representing a large integer. Return the largest-valued odd integer (as a string) that is a non-empty substring of num, or an empty string "" if no odd integer exists. A substring is a contiguous sequence of characters within a string.   Example 1: Input: num = "52" Output: "5" Explanation: The only non-empty substrings are "5", "2", and "52". "5" is the only odd number.

Example 2: Input: num = "4206" Output: "" Explanation: There are no odd numbers in "4206".

Example 3: Input: num = "35427" Output: "35427" Explanation: "35427" is already an odd number.

Constraints:

1 <= num.length <= 105 num only consists of digits and does not contain any leading zeros.

==

Problem Number: 357 URL: https://leetcode.com/problems/remove-one-element-to-make-the-array-strictly-increasing Title: 1909. Remove One Element to Make the Array Strictly Increasing Problem Description: Given a 0-indexed integer array nums, return true if it can be made strictly increasing after removing exactly one element, or false otherwise. If the array is already strictly increasing, return true. The array nums is strictly increasing if nums[i - 1] < nums[i] for each index (1 <= i < nums.length). Example 1: Input: nums = [1,2,10,5,7] Output: true Explanation: By removing 10 at index 2 from nums, it becomes [1,2,5,7]. [1,2,5,7] is strictly increasing, so return true.

Example 2: Input: nums = [2,3,1,2] Output: false Explanation: [3,1,2] is the result of removing the element at index 0. [2,1,2] is the result of removing the element at index 1. [2,3,2] is the result of removing the element at index 2. [2,3,1] is the result of removing the element at index 3. No resulting array is strictly increasing, so return false. Example 3: Input: nums = [1,1,1] Output: false Explanation: The result of removing any element is [1,1]. [1,1] is not strictly increasing, so return false.

Constraints:

2 <= nums.length <= 1000 1 <= nums[i] <= 1000

==

Problem Number: 358 URL: https://leetcode.com/problems/maximum-product-difference-between-two-pairs Title: 1913. Maximum Product Difference Between Two Pairs Problem Description: The product difference between two pairs (a, b) and (c, d) is defined as (a * b) - (c * d).

For example, the product difference between (5, 6) and (2, 7) is (5 * 6) - (2 * 7) = 16.

Given an integer array nums, choose four distinct indices w, x, y, and z such that the product difference between pairs (nums[w], nums[x]) and (nums[y], nums[z]) is maximized. Return the maximum such product difference. Example 1: Input: nums = [5,6,2,7,4] Output: 34 Explanation: We can choose indices 1 and 3 for the first pair (6, 7) and indices 2 and 4 for the second pair (2, 4). The product difference is (6 * 7) - (2 * 4) = 34.

Example 2: Input: nums = [4,2,5,9,7,4,8] Output: 64 Explanation: We can choose indices 3 and 6 for the first pair (9, 8) and indices 1 and 5 for the second pair (2, 4). The product difference is (9 * 8) - (2 * 4) = 64.

Constraints:

4 <= nums.length <= 104 1 <= nums[i] <= 104

==

Problem Number: 359 URL: https://leetcode.com/problems/build-array-from-permutation Title: 1920. Build Array from Permutation Problem Description: Given a zero-based permutation nums (0-indexed), build an array ans of the same length where ans[i] = nums[nums[i]] for each 0 <= i < nums.length and return it. A zero-based permutation nums is an array of distinct integers from 0 to nums.length - 1 (inclusive). Example 1: Input: nums = [0,2,1,5,3,4] Output: [0,1,2,4,5,3] Explanation: The array ans is built as follows: ans = [nums[nums[0]], nums[nums[1]], nums[nums[2]], nums[nums[3]], nums[nums[4]], nums[nums[5]]] = [nums[0], nums[2], nums[1], nums[5], nums[3], nums[4]] = [0,1,2,4,5,3] Example 2: Input: nums = [5,0,1,2,3,4] Output: [4,5,0,1,2,3] Explanation: The array ans is built as follows: ans = [nums[nums[0]], nums[nums[1]], nums[nums[2]], nums[nums[3]], nums[nums[4]], nums[nums[5]]] = [nums[5], nums[0], nums[1], nums[2], nums[3], nums[4]] = [4,5,0,1,2,3] Constraints:

1 <= nums.length <= 1000 0 <= nums[i] < nums.length The elements in nums are distinct.

Follow-up: Can you solve it without using an extra space (i.e., O(1) memory)?

==

Problem Number: 360 URL: https://leetcode.com/problems/count-square-sum-triples Title: 1925. Count Square Sum Triples Problem Description: A square triple (a,b,c) is a triple where a, b, and c are integers and a2 + b2 = c2. Given an integer n, return the number of square triples such that 1 <= a, b, c <= n. Example 1: Input: n = 5 Output: 2 Explanation: The square triples are (3,4,5) and (4,3,5).

Example 2: Input: n = 10 Output: 4 Explanation: The square triples are (3,4,5), (4,3,5), (6,8,10), and (8,6,10).

Constraints:

1 <= n <= 250

==

Problem Number: 361 URL: https://leetcode.com/problems/concatenation-of-array Title: 1929. Concatenation of Array Problem Description: Given an integer array nums of length n, you want to create an array ans of length 2n where ans[i] == nums[i] and ans[i + n] == nums[i] for 0 <= i < n (0-indexed). Specifically, ans is the concatenation of two nums arrays. Return the array ans. Example 1: Input: nums = [1,2,1] Output: [1,2,1,1,2,1] Explanation: The array ans is formed as follows: - ans = [nums[0],nums[1],nums[2],nums[0],nums[1],nums[2]] - ans = [1,2,1,1,2,1] Example 2: Input: nums = [1,3,2,1] Output: [1,3,2,1,1,3,2,1] Explanation: The array ans is formed as follows: - ans = [nums[0],nums[1],nums[2],nums[3],nums[0],nums[1],nums[2],nums[3]] - ans = [1,3,2,1,1,3,2,1]

Constraints:

n == nums.length 1 <= n <= 1000 1 <= nums[i] <= 1000

==
Problem Number: 362 URL: https://leetcode.com/problems/maximum-number-of-words-you-can-type Title: 1935. Maximum Number of Words You Can Type Problem Description: There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly. Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard. Example 1: Input: text = "hello world", brokenLetters = "ad" Output: 1 Explanation: We cannot type "world" because the 'd' key is broken.

Example 2: Input: text = "leet code", brokenLetters = "lt" Output: 1 Explanation: We cannot type "leet" because the 'l' and 't' keys are broken.

Example 3: Input: text = "leet code", brokenLetters = "e" Output: 0 Explanation: We cannot type either word because the 'e' key is broken.

Constraints:

1 <= text.length <= 104 0 <= brokenLetters.length <= 26 text consists of words separated by a single space without any leading or trailing spaces. Each word only consists of lowercase English letters. brokenLetters consists of distinct lowercase English letters.

==
Problem Number: 363 URL: https://leetcode.com/problems/check-if-all-characters-have-equal-number-of-occurrences Title: 1941. Check if All Characters Have Equal Number of Occurrences Problem Description: Given a string s, return true if s is a good string, or false otherwise. A string s is good if all the characters that appear in s have the same number of occurrences (i.e., the same frequency). Example 1: Input: s = "abacbc" Output: true Explanation: The characters that appear in s are 'a', 'b', and 'c'. All characters occur 2 times in s.

Example 2: Input: s = "aaabb" Output: false Explanation: The characters that appear in s are 'a' and 'b'. 'a' occurs 3 times while 'b' occurs 2 times, which is not the same number of times.

Constraints:

1 <= s.length <= 1000 s consists of lowercase English letters.

==
Problem Number: 364 URL: https://leetcode.com/problems/sum-of-digits-of-string-after-convert Title: 1945. Sum of Digits of String After Convert Problem Description: You are given a string s consisting of lowercase English letters, and an integer k. First, convert s into an integer by replacing each letter with its position in the alphabet (i.e., replace 'a' with 1, 'b' with 2, ..., 'z' with 26). Then, transform the integer by replacing it with the sum of its digits. Repeat

the transform operation k times in total. For example, if s = "zbax" and k = 2, then the resulting integer would be 8 by the following operations:

Convert: "zbax" "(26)(2)(1)(24)" "262124" 262124 Transform #1: 262124 2 + 6 + 2 + 1 + 2 + 4 17 Transform #2: 17 1 + 7 8

Return the resulting integer after performing the operations described above. Example 1: Input: s = "iiii", k = 1 Output: 36 Explanation: The operations are as follows: - Convert: "iiii" "(9)(9)(9)(9)" "9999" 9999 - Transform #1: 9999 9 + 9 + 9 + 9 36 Thus the resulting integer is 36.

Example 2: Input: s = "leetcode", k = 2 Output: 6 Explanation: The operations are as follows: - Convert: "leetcode" "(12)(5)(5)(20)(3)(15)(4)(5)" "12552031545" 12552031545 - Transform #1: 12552031545 1 + 2 + 5 + 5 + 2 + 0 + 3 + 1 + 5 + 4 + 5 33 - Transform #2: 33 3 + 3 6 Thus the resulting integer is 6.

Example 3: Input: s = "zbax", k = 2 Output: 8

Constraints:

1 <= s.length <= 100 1 <= k <= 10 s consists of lowercase English letters.

==
Problem Number: 365 URL: https://leetcode.com/problems/three-divisors Title: 1952. Three Divisors Problem Description: Given an integer n, return true if n has exactly three positive divisors. Otherwise, return false. An integer m is a divisor of n if there exists an integer k such that n = k * m. Example 1: Input: n = 2 Output: false Explantion: 2 has only two divisors: 1 and 2.

Example 2: Input: n = 4 Output: true Explantion: 4 has three divisors: 1, 2, and 4.

Constraints:

1 <= n <= 104

==
Problem Number: 366 URL: https://leetcode.com/problems/delete-characters-to-make-fancy-string Title: 1957. Delete Characters to Make Fancy String Problem Description: A fancy string is a string where no three consecutive characters are equal. Given a string s, delete the minimum possible number of characters from s to make it fancy. Return the final string after the deletion. It can be shown that the answer will always be unique. Example 1: Input: s = "leeetcode" Output: "leetcode" Explanation: Remove an 'e' from the first group of 'e's to create "leetcode". No three consecutive characters are equal, so return "leetcode".

Example 2: Input: s = "aaabaaaa" Output: "aabaa" Explanation: Remove an 'a' from the first group of 'a's to create "aabaaaa". Remove two 'a's from the second group of 'a's to create "aabaa". No three consecutive characters are equal, so return "aabaa".

Example 3: Input: s = ”aab” Output: ”aab” Explanation: No three consecutive characters are equal, so return ”aab”.

Constraints:

1 <= s.length <= 105 s consists only of lowercase English letters.

==

Problem Number: 367 URL: https://leetcode.com/problems/check-if-string-is-a-prefix-of-array Title: 1961. Check If String Is a Prefix of Array Problem Description: Given a string s and an array of strings words, determine whether s is a prefix string of words. A string s is a prefix string of words if s can be made by concatenating the first k strings in words for some positive k no larger than words.length. Return true if s is a prefix string of words, or false otherwise. Example 1: Input: s = ”iloveleetcode”, words = [”i”,”love”,”leetcode”,”apples”] Output: true Explanation: s can be made by concatenating ”i”, ”love”, and ”leetcode” together.

Example 2: Input: s = ”iloveleetcode”, words = [”apples”,”i”,”love”,”leetcode”] Output: false Explanation: It is impossible to make s using a prefix of arr. Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 20 1 <= s.length <= 1000 words[i] and s consist of only lowercase English letters.

==

Problem Number: 368 URL: https://leetcode.com/problems/number-of-strings-that-appear-as-substrings-in-word Title: 1967. Number of Strings That Appear as Substrings in Word Problem Description: Given an array of strings patterns and a string word, return the number of strings in patterns that exist as a substring in word. A substring is a contiguous sequence of characters within a string. Example 1: Input: patterns = [”a”,”abc”,”bc”,”d”], word = ”abc” Output: 3 Explanation: - ”a” appears as a substring in ”abc”. - ”abc” appears as a substring in ”abc”. - ”bc” appears as a substring in ”abc”. - ”d” does not appear as a substring in ”abc”. 3 of the strings in patterns appear as a substring in word.

Example 2: Input: patterns = [”a”,”b”,”c”], word = ”aaaaabbbbb” Output: 2 Explanation: - ”a” appears as a substring in ”aaaaabbbbb”. - ”b” appears as a substring in ”aaaaabbbbb”. - ”c” does not appear as a substring in ”aaaaabbbbb”. 2 of the strings in patterns appear as a substring in word.

Example 3: Input: patterns = [”a”,”a”,”a”], word = ”ab” Output: 3 Explanation: Each of the patterns appears as a substring in word ”ab”.

Constraints:

1 <= patterns.length <= 100 1 <= patterns[i].length <= 100 1 <= word.length <= 100 patterns[i] and word consist of lowercase English letters.

==

Problem Number: 369 URL: https://leetcode.com/problems/find-if-path-exists-in-graph Title: 1971. Find if Path Exists in Graph Problem Description: There is a bi-directional graph with n vertices, where each vertex is labeled from 0 to n - 1 (inclusive). The edges in the graph are represented as a 2D integer array edges, where each edges[i] = [ui, vi] denotes a bi-directional edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself. You want to determine if there is a valid path that exists from vertex source to vertex destination. Given edges and the integers n, source, and destination, return true if there is a valid path from source to destination, or false otherwise. Example 1:

Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2 Output: true Explanation: There are two paths from vertex 0 to vertex 2: - 0 → 1 → 2 - 0 → 2

Example 2:

Input: n = 6, edges = [[0,1],[0,2],[3,5],[5,4],[4,3]], source = 0, destination = 5 Output: false Explanation: There is no path from vertex 0 to vertex 5.

Constraints:

1 <= n <= 2 * 105 0 <= edges.length <= 2 * 105 edges[i].length == 2 0 <= ui, vi <= n - 1 ui != vi 0 <= source, destination <= n - 1 There are no duplicate edges. There are no self edges.

==
Problem Number: 370 URL: https://leetcode.com/problems/minimum-time-to-type-word-using-special-typewriter Title: 1974. Minimum Time to Type Word Using Special Typewriter Problem Description: There is a special typewriter with lowercase English letters 'a' to 'z' arranged in a circle with a pointer. A character can only be typed if the pointer is pointing to that character. The pointer is initially pointing to the character 'a'.

Each second, you may perform one of the following operations:

Move the pointer one character counterclockwise or clockwise. Type the character the pointer is currently on.

Given a string word, return the minimum number of seconds to type out the characters in word. Example 1: Input: word = "abc" Output: 5 Explanation: The characters are printed as follows: - Type the character 'a' in 1 second since the pointer is initially on 'a'. - Move the pointer clockwise to 'b' in 1 second. - Type the character 'b' in 1 second. - Move the pointer clockwise to 'c' in 1 second. - Type the character 'c' in 1 second.

Example 2: Input: word = "bza" Output: 7 Explanation: The characters are printed as follows: - Move the pointer clockwise to 'b' in 1 second. - Type the character 'b' in 1 second. - Move the pointer counterclockwise to 'z' in 2 seconds. - Type the character 'z' in 1 second. - Move the pointer clockwise to 'a' in 1 second. - Type the character 'a' in 1 second.

Example 3: Input: word = ”zjpc” Output: 34 Explanation: The characters are printed as follows: - Move the pointer counterclockwise to ’z’ in 1 second. - Type the character ’z’ in 1 second. - Move the pointer clockwise to ’j’ in 10 seconds. - Type the character ’j’ in 1 second. - Move the pointer clockwise to ’p’ in 6 seconds. - Type the character ’p’ in 1 second. - Move the pointer counterclockwise to ’c’ in 13 seconds. - Type the character ’c’ in 1 second.

Constraints:

1 <= word.length <= 100 word consists of lowercase English letters.

==
Problem Number: 371 URL: https://leetcode.com/problems/find-greatest-common-divisor-of-array Title: 1979. Find Greatest Common Divisor of Array Problem Description: Given an integer array nums, return the greatest common divisor of the smallest number and largest number in nums. The greatest common divisor of two numbers is the largest positive integer that evenly divides both numbers. Example 1: Input: nums = [2,5,6,9,10] Output: 2 Explanation: The smallest number in nums is 2. The largest number in nums is 10. The greatest common divisor of 2 and 10 is 2.

Example 2: Input: nums = [7,5,6,8,3] Output: 1 Explanation: The smallest number in nums is 3. The largest number in nums is 8. The greatest common divisor of 3 and 8 is 1.

Example 3: Input: nums = [3,3] Output: 3 Explanation: The smallest number in nums is 3. The largest number in nums is 3. The greatest common divisor of 3 and 3 is 3.

Constraints:

2 <= nums.length <= 1000 1 <= nums[i] <= 1000

==
Problem Number: 372 URL: https://leetcode.com/problems/minimum-difference-between-highest-and-lowest-of-k-scores Title: 1984. Minimum Difference Between Highest and Lowest of K Scores Problem Description: You are given a 0-indexed integer array nums, where nums[i] represents the score of the ith student. You are also given an integer k. Pick the scores of any k students from the array so that the difference between the highest and the lowest of the k scores is minimized. Return the minimum possible difference. Example 1: Input: nums = [90], k = 1 Output: 0 Explanation: There is one way to pick score(s) of one student: - [90]. The difference between the highest and lowest score is 90 - 90 = 0. The minimum possible difference is 0.

Example 2: Input: nums = [9,4,1,7], k = 2 Output: 2 Explanation: There are six ways to pick score(s) of two students: - [9,4,1,7]. The difference between the highest and lowest score is 9 - 4 = 5. - [9,4,1,7]. The difference between the highest and lowest score is 9 - 1 = 8. - [9,4,1,7]. The difference between the highest and lowest score is 9 - 7 = 2. - [9,4,1,7]. The difference between the highest and lowest score is 4 - 1 = 3. - [9,4,1,7]. The difference between the highest and lowest score is 7 - 4 = 3. - [9,4,1,7]. The difference between the highest and lowest score is 7 - 1 = 6. The minimum possible difference is 2. Constraints:

1 <= k <= nums.length <= 1000 0 <= nums[i] <= 105

==
Problem Number: 373 URL: https://leetcode.com/problems/find-the-middle-index-in-array Title: 1991. Find the Middle Index in Array Problem Description: Given a 0-indexed integer array nums, find the leftmost middleIndex (i.e., the smallest amongst all the possible ones). A middleIndex is an index where nums[0] + nums[1] + ... + nums[middleIndex-1] == nums[middleIndex+1] + nums[middleIndex+2] + ... + nums[nums.length-1]. If middleIndex == 0, the left side sum is considered to be 0. Similarly, if middleIndex == nums.length - 1, the right side sum is considered to be 0. Return the leftmost middleIndex that satisfies the condition, or -1 if there is no such index. Example 1: Input: nums = [2,3,-1,8,4] Output: 3 Explanation: The sum of the numbers before index 3 is: 2 + 3 + -1 = 4 The sum of the numbers after index 3 is: 4 = 4

Example 2: Input: nums = [1,-1,4] Output: 2 Explanation: The sum of the numbers before index 2 is: 1 + -1 = 0 The sum of the numbers after index 2 is: 0

Example 3: Input: nums = [2,5] Output: -1 Explanation: There is no valid middleIndex.

Constraints:

1 <= nums.length <= 100 -1000 <= nums[i] <= 1000

Note: This question is the same as 724: https://leetcode.com/problems/find-pivot-index/

==
Problem Number: 374 URL: https://leetcode.com/problems/count-special-quadruplets Title: 1995. Count Special Quadruplets Problem Description: Given a 0-indexed integer array nums, return the number of distinct quadruplets (a, b, c, d) such that:

nums[a] + nums[b] + nums[c] == nums[d], and a < b < c < d

Example 1: Input: nums = [1,2,3,6] Output: 1 Explanation: The only quadruplet that satisfies the requirement is (0, 1, 2, 3) because 1 + 2 + 3 == 6.

Example 2: Input: nums = [3,3,6,4,5] Output: 0 Explanation: There are no such quadruplets in [3,3,6,4,5].

Example 3: Input: nums = [1,1,1,3,5] Output: 4 Explanation: The 4 quadruplets that satisfy the requirement are: - (0, 1, 2, 3): 1 + 1 + 1 == 3 - (0, 1, 3, 4): 1 + 1 + 3 == 5 - (0, 2, 3, 4): 1 + 1 + 3 == 5 - (1, 2, 3, 4): 1 + 1 + 3 == 5

Constraints:

4 <= nums.length <= 50 1 <= nums[i] <= 100

==

Problem Number: 375 URL: https://leetcode.com/problems/reverse-prefix-of-word Title: 2000. Reverse Prefix of Word Problem Description: Given a 0-indexed string word and a character ch, reverse the segment of word that starts at index 0 and ends at the index of the first occurrence of ch (inclusive). If the character ch does not exist in word, do nothing.

For example, if word = "abcdefd" and ch = "d", then you should reverse the segment that starts at 0 and ends at 3 (inclusive). The resulting string will be "dcbaefd".

Return the resulting string.   Example 1: Input: word = "abcdefd", ch = "d" Output: "dcbaefd" Explanation: The first occurrence of "d" is at index 3. Reverse the part of word from 0 to 3 (inclusive), the resulting string is "dcbaefd".

Example 2: Input: word = "xyxzxe", ch = "z" Output: "zxyxxe" Explanation: The first and only occurrence of "z" is at index 3. Reverse the part of word from 0 to 3 (inclusive), the resulting string is "zxyxxe".

Example 3: Input: word = "abcd", ch = "z" Output: "abcd" Explanation: "z" does not exist in word. You should not do any reverse operation, the resulting string is "abcd".

Constraints:

1 <= word.length <= 250 word consists of lowercase English letters. ch is a lowercase English letter.

==

Problem Number: 376 URL: https://leetcode.com/problems/count-number-of-pairs-with-absolute-difference-k Title: 2006. Count Number of Pairs With Absolute Difference K Problem Description: Given an integer array nums and an integer k, return the number of pairs (i, j) where i < j such that |nums[i] - nums[j]| == k. The value of |x| is defined as:

x if x >= 0. -x if x < 0.

Example 1: Input: nums = [1,2,2,1], k = 1 Output: 4 Explanation: The pairs with an absolute difference of 1 are: - [1,2,2,1] - [1,2,2,1] - [1,2,2,1] - [1,2,2,1]

Example 2: Input: nums = [1,3], k = 3 Output: 0 Explanation: There are no pairs with an absolute difference of 3.

Example 3: Input: nums = [3,2,1,5,4], k = 2 Output: 3 Explanation: The pairs with an absolute difference of 2 are: - [3,2,1,5,4] - [3,2,1,5,4] - [3,2,1,5,4]

Constraints:

1 <= nums.length <= 200 1 <= nums[i] <= 100 1 <= k <= 99

==
Problem Number: 377 URL: https://leetcode.com/problems/final-value-of-variable-after-performing-operations Title: 2011. Final Value of Variable After Performing Operations Problem Description: There is a programming language with only four operations and one variable X:

++X and X++ increments the value of the variable X by 1. --X and X-- decrements the value of the variable X by 1.

Initially, the value of X is 0. Given an array of strings operations containing a list of operations, return the final value of X after performing all the operations. Example 1: Input: operations = ["--X","X++","X++"] Output: 1 Explanation: The operations are performed as follows: Initially, X = 0. --X: X is decremented by 1, X = 0 - 1 = -1. X++: X is incremented by 1, X = -1 + 1 = 0. X++: X is incremented by 1, X = 0 + 1 = 1.

Example 2: Input: operations = ["++X","++X","X++"] Output: 3 Explanation: The operations are performed as follows: Initially, X = 0. ++X: X is incremented by 1, X = 0 + 1 = 1. ++X: X is incremented by 1, X = 1 + 1 = 2. X++: X is incremented by 1, X = 2 + 1 = 3.

Example 3: Input: operations = ["X++","++X","--X","X--"] Output: 0 Explanation: The operations are performed as follows: Initially, X = 0. X++: X is incremented by 1, X = 0 + 1 = 1. ++X: X is incremented by 1, X = 1 + 1 = 2. --X: X is decremented by 1, X = 2 - 1 = 1. X--: X is decremented by 1, X = 1 - 1 = 0.

Constraints:

1 <= operations.length <= 100 operations[i] will be either "++X", "X++", "--X", or "X--".

==
Problem Number: 378 URL: https://leetcode.com/problems/maximum-difference-between-increasing-elements Title: 2016. Maximum Difference Between Increasing Elements Problem Description: Given a 0-indexed integer array nums of size n, find the maximum difference between nums[i] and nums[j] (i.e., nums[j] - nums[i]), such that 0 <= i < j < n and nums[i] < nums[j]. Return the maximum difference. If no such i and j exists, return -1. Example 1: Input: nums = [7,1,5,4] Output: 4 Explanation: The maximum difference occurs with i = 1 and j = 2, nums[j] - nums[i] = 5 - 1 = 4. Note that with i = 1 and j = 0, the difference nums[j] - nums[i] = 7 - 1 = 6, but i > j, so it is not valid.

Example 2: Input: nums = [9,4,3,2] Output: -1 Explanation: There is no i and j such that i < j and nums[i] < nums[j].

Example 3: Input: nums = [1,5,2,10] Output: 9 Explanation: The maximum difference occurs with i = 0 and j = 3, nums[j] - nums[i] = 10 - 1 = 9.

Constraints:

n == nums.length 2 <= n <= 1000 1 <= nums[i] <= 109

==

Problem Number: 379 URL: https://leetcode.com/problems/convert-1d-array-into-2d-array Title: 2022. Convert 1D Array Into 2D Array Problem Description: You are given a 0-indexed 1-dimensional (1D) integer array original, and two integers, m and n. You are tasked with creating a 2-dimensional (2D) array with m rows and n columns using all the elements from original. The elements from indices 0 to n - 1 (inclusive) of original should form the first row of the constructed 2D array, the elements from indices n to 2 * n - 1 (inclusive) should form the second row of the constructed 2D array, and so on. Return an m x n 2D array constructed according to the above procedure, or an empty 2D array if it is impossible. Example 1:

Input: original = [1,2,3,4], m = 2, n = 2 Output: [[1,2],[3,4]] Explanation: The constructed 2D array should contain 2 rows and 2 columns. The first group of n=2 elements in original, [1,2], becomes the first row in the constructed 2D array. The second group of n=2 elements in original, [3,4], becomes the second row in the constructed 2D array.

Example 2: Input: original = [1,2,3], m = 1, n = 3 Output: [[1,2,3]] Explanation: The constructed 2D array should contain 1 row and 3 columns. Put all three elements in original into the first row of the constructed 2D array.

Example 3: Input: original = [1,2], m = 1, n = 1 Output: [] Explanation: There are 2 elements in original. It is impossible to fit 2 elements in a 1x1 2D array, so return an empty 2D array.

Constraints:

1 <= original.length <= 5 * 104 1 <= original[i] <= 105 1 <= m, n <= 4 * 104

==

Problem Number: 380 URL: https://leetcode.com/problems/minimum-moves-to-convert-string Title: 2027. Minimum Moves to Convert String Problem Description: You are given a string s consisting of n characters which are either 'X' or 'O'. A move is defined as selecting three consecutive characters of s and converting them to 'O'. Note that if a move is applied to the character 'O', it will stay the same. Return the minimum number of moves required so that all the characters of s are converted to 'O'. Example 1: Input: s = "XXX" Output: 1 Explanation: XXX -> OOO We select all the 3 characters and convert them in one move.

Example 2: Input: s = "XXOX" Output: 2 Explanation: XXOX -> OOOX -> OOOO We select the first 3 characters in the first move, and convert them to 'O'. Then we select the last 3 characters and convert them so that the final string contains all 'O's. Example 3: Input: s = "OOOO" Output: 0 Explanation: There are no 'X's in s to convert.

Constraints:

3 <= s.length <= 1000 s[i] is either 'X' or 'O'.

===
Problem Number: 381 URL: https://leetcode.com/problems/two-out-of-three Title: 2032. Two Out of Three Problem Description: Given three integer arrays nums1, nums2, and nums3, return a distinct array containing all the values that are present in at least two out of the three arrays. You may return the values in any order.   Example 1: Input: nums1 = [1,1,3,2], nums2 = [2,3], nums3 = [3] Output: [3,2] Explanation: The values that are present in at least two arrays are: - 3, in all three arrays. - 2, in nums1 and nums2.

Example 2: Input: nums1 = [3,1], nums2 = [2,3], nums3 = [1,2] Output: [2,3,1] Explanation: The values that are present in at least two arrays are: - 2, in nums2 and nums3. - 3, in nums1 and nums2. - 1, in nums1 and nums3.

Example 3: Input: nums1 = [1,2,2], nums2 = [4,3,3], nums3 = [5] Output: [] Explanation: No value is present in at least two arrays.

Constraints:

1 <= nums1.length, nums2.length, nums3.length <= 100 1 <= nums1[i], nums2[j], nums3[k] <= 100

===
Problem Number: 382 URL: https://leetcode.com/problems/minimum-number-of-moves-to-seat-everyone Title: 2037. Minimum Number of Moves to Seat Everyone Problem Description: There are n seats and n students in a room. You are given an array seats of length n, where seats[i] is the position of the ith seat. You are also given the array students of length n, where students[j] is the position of the jth student. You may perform the following move any number of times:

Increase or decrease the position of the ith student by 1 (i.e., moving the ith student from position x to x + 1 or x - 1)

Return the minimum number of moves required to move each student to a seat such that no two students are in the same seat. Note that there may be multiple seats or students in the same position at the beginning.   Example 1: Input: seats = [3,1,5], students = [2,7,4] Output: 4 Explanation: The students are moved as follows: - The first student is moved from from position 2 to position 1 using 1 move. - The second student is moved from from position 7 to position 5 using 2 moves. - The third student is moved from from position 4 to position 3 using 1 move. In total, 1 + 2 + 1 = 4 moves were used.

Example 2: Input: seats = [4,1,5,9], students = [1,3,2,6] Output: 7 Explanation: The students are moved as follows: - The first student is not moved. - The second student is moved from from position 3 to position 4 using 1 move. - The third student is moved from from position 2 to position 5 using 3 moves. - The fourth student is moved from from position 6 to position 9 using 3 moves. In total, 0 + 1 + 3 + 3 = 7 moves were used.

Example 3: Input: seats = [2,2,6,6], students = [1,3,2,6] Output: 4 Explanation: Note that there are two seats at position 2 and two seats at position 6. The students are moved as follows: - The first student is moved from from position 1 to position 2 using 1 move. - The second student is moved from from position 3 to position 6 using 3 moves. - The third student is not moved. - The fourth student is not moved. In total, 1 + 3 + 0 + 0 = 4 moves were used.

Constraints:

n == seats.length == students.length 1 <= n <= 100 1 <= seats[i], students[j] <= 100

==
Problem Number: 383 URL: https://leetcode.com/problems/check-if-numbers-are-ascending-in-a-sentence Title: 2042. Check if Numbers Are Ascending in a Sentence Problem Description: A sentence is a list of tokens separated by a single space with no leading or trailing spaces. Every token is either a positive number consisting of digits 0-9 with no leading zeros, or a word consisting of lowercase English letters.

For example, ”a puppy has 2 eyes 4 legs” is a sentence with seven tokens: ”2” and ”4” are numbers and the other tokens such as ”puppy” are words.

Given a string s representing a sentence, you need to check if all the numbers in s are strictly increasing from left to right (i.e., other than the last number, each number is strictly smaller than the number on its right in s). Return true if so, or false otherwise. Example 1:

Input: s = ”1 box has 3 blue 4 red 6 green and 12 yellow marbles” Output: true Explanation: The numbers in s are: 1, 3, 4, 6, 12. They are strictly increasing from left to right: 1 < 3 < 4 < 6 < 12.

Example 2: Input: s = ”hello world 5 x 5” Output: false Explanation: The numbers in s are: 5, 5. They are not strictly increasing.

Example 3:

Input: s = ”sunset is at 7 51 pm overnight lows will be in the low 50 and 60 s” Output: false Explanation: The numbers in s are: 7, 51, 50, 60. They are not strictly increasing.

Constraints:

3 <= s.length <= 200 s consists of lowercase English letters, spaces, and digits from 0 to 9, inclusive. The number of tokens in s is between 2 and 100, inclusive. The tokens in s are separated by a single space. There are at least two numbers in s. Each number in s is a positive number less than 100, with no leading zeros. s contains no leading or trailing spaces.

==
Problem Number: 384 URL: https://leetcode.com/problems/number-of-valid-words-in-a-sentence Title: 2047. Number of Valid Words in a Sentence Problem Description: A sentence consists of lowercase letters ('a' to 'z'), digits ('0' to '9'), hyphens ('-'), punctuation marks ('!', '.', and ','), and spaces (' ') only. Each sentence can be broken down into one or more tokens separated by one or more spaces ' '. A token is a valid word if all three of the following are true:

It only contains lowercase letters, hyphens, and/or punctuation (no digits). There is at most one hyphen '-'. If present, it must be surrounded by lowercase characters ("a-b" is valid, but "-ab" and "ab-" are not valid). There is at most one punctuation mark. If present, it must be at the end of the token ("ab,", "cd!", and "." are valid, but "a!b" and "c.," are not valid).

Examples of valid words include "a-b.", "afad", "ba-c", "a!", and "!". Given a string sentence, return the number of valid words in sentence. Example 1: Input: sentence = "cat and dog" Output: 3 Explanation: The valid words in the sentence are "cat", "and", and "dog".

Example 2: Input: sentence = "!this 1-s b8d!" Output: 0 Explanation: There are no valid words in the sentence. "!this" is invalid because it starts with a punctuation mark. "1-s" and "b8d" are invalid because they contain digits.

Example 3: Input: sentence = "alice and bob are playing stone-game10" Output: 5 Explanation: The valid words in the sentence are "alice", "and", "bob", "are", and "playing". "stone-game10" is invalid because it contains digits.

Constraints:

1 <= sentence.length <= 1000 sentence only contains lowercase English letters, digits, ' ', '-', '!', '.', and ','. There will be at least 1 token.

==
Problem Number: 385 URL: https://leetcode.com/problems/kth-distinct-string-in-an-array Title: 2053. Kth Distinct String in an Array Problem Description: A distinct string is a string that is present only once in an array. Given an array of strings arr, and an integer k, return the kth distinct string present in arr. If there are fewer than k distinct strings, return an empty string "". Note that the strings are considered in the order in which they appear in the array. Example 1: Input: arr = ["d","b","c","b","c","a"], k = 2 Output: "a" Explanation: The only distinct strings in arr are "d" and "a". "d" appears 1st, so it is the 1st distinct string. "a" appears 2nd, so it is the 2nd distinct string. Since k == 2, "a" is returned.

Example 2: Input: arr = ["aaa","aa","a"], k = 1 Output: "aaa" Explanation: All strings in arr are distinct, so the 1st string "aaa" is returned.

Example 3: Input: arr = ["a","b","a"], k = 3 Output: "" Explanation: The only distinct string is "b". Since there are fewer than 3 distinct strings, we return an empty string "".

Constraints:

1 <= k <= arr.length <= 1000 1 <= arr[i].length <= 5 arr[i] consists of lowercase English letters.

==
Problem Number: 386 URL: https://leetcode.com/problems/smallest-index-with-equal-value Title: 2057. Smallest Index With Equal Value Problem Description: Given a 0-indexed integer array nums, return the smallest index i of nums such that i mod 10 == nums[i], or -1 if such index does not exist. x mod y denotes the remainder when x is divided by y. Example 1: Input: nums = [0,1,2] Output: 0 Explanation: i=0: 0 mod 10 = 0 == nums[0]. i=1: 1 mod 10 = 1 == nums[1]. i=2: 2 mod 10 = 2 == nums[2]. All indices have i mod 10 == nums[i], so we return the smallest index 0.

Example 2: Input: nums = [4,3,2,1] Output: 2 Explanation: i=0: 0 mod 10 = 0 != nums[0]. i=1: 1 mod 10 = 1 != nums[1]. i=2: 2 mod 10 = 2 == nums[2]. i=3: 3 mod 10 = 3 != nums[3]. 2 is the only index which has i mod 10 == nums[i].

Example 3: Input: nums = [1,2,3,4,5,6,7,8,9,0] Output: -1 Explanation: No index satisfies i mod 10 == nums[i].

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 9

==
Problem Number: 387 URL: https://leetcode.com/problems/count-vowel-substrings-of-a-string Title: 2062. Count Vowel Substrings of a String Problem Description: A substring is a contiguous (non-empty) sequence of characters within a string. A vowel substring is a substring that only consists of vowels ('a', 'e', 'i', 'o', and 'u') and has all five vowels present in it. Given a string word, return the number of vowel substrings in word. Example 1: Input: word = "aeiouu" Output: 2 Explanation: The vowel substrings of word are as follows (underlined): - "aeiouu" - "aeiouu"

Example 2: Input: word = "unicornarihan" Output: 0 Explanation: Not all 5 vowels are present, so there are no vowel substrings.

Example 3: Input: word = "cuaieuouac" Output: 7 Explanation: The vowel substrings of word are as follows (underlined): - "cuaieuouac" - "cuaieuouac" - "cuaieuouac" - "cuaieuouac" - "cuaieuouac" - "cuaieuouac" - "cuaieuouac"

Constraints:

1 <= word.length <= 100 word consists of lowercase English letters only.

==
Problem Number: 388 URL: https://leetcode.com/problems/check-whether-two-strings-are-almost-equivalent Title: 2068. Check Whether Two Strings are Almost Equivalent Problem Description: Two strings word1 and word2 are

considered almost equivalent if the differences between the frequencies of each letter from 'a' to 'z' between word1 and word2 is at most 3. Given two strings word1 and word2, each of length n, return true if word1 and word2 are almost equivalent, or false otherwise. The frequency of a letter x is the number of times it occurs in the string.   Example 1: Input: word1 = "aaaa", word2 = "bccb" Output: false Explanation: There are 4 'a's in "aaaa" but 0 'a's in "bccb". The difference is 4, which is more than the allowed 3.

Example 2: Input: word1 = "abcdeef", word2 = "abaaacc" Output: true Explanation: The differences between the frequencies of each letter in word1 and word2 are at most 3: - 'a' appears 1 time in word1 and 4 times in word2. The difference is 3. - 'b' appears 1 time in word1 and 1 time in word2. The difference is 0. - 'c' appears 1 time in word1 and 2 times in word2. The difference is 1. - 'd' appears 1 time in word1 and 0 times in word2. The difference is 1. - 'e' appears 2 times in word1 and 0 times in word2. The difference is 2. - 'f' appears 1 time in word1 and 0 times in word2. The difference is 1.

Example 3: Input: word1 = "cccddabba", word2 = "babababab" Output: true Explanation: The differences between the frequencies of each letter in word1 and word2 are at most 3: - 'a' appears 2 times in word1 and 4 times in word2. The difference is 2. - 'b' appears 2 times in word1 and 5 times in word2. The difference is 3. - 'c' appears 3 times in word1 and 0 times in word2. The difference is 3. - 'd' appears 2 times in word1 and 0 times in word2. The difference is 2.

Constraints:

n == word1.length == word2.length 1 <= n <= 100 word1 and word2 consist only of lowercase English letters.

==
Problem Number: 389 URL: https://leetcode.com/problems/time-needed-to-buy-tickets Title: 2073. Time Needed to Buy Tickets Problem Description: There are n people in a line queuing to buy tickets, where the 0th person is at the front of the line and the (n - 1)th person is at the back of the line. You are given a 0-indexed integer array tickets of length n where the number of tickets that the ith person would like to buy is tickets[i]. Each person takes exactly 1 second to buy a ticket. A person can only buy 1 ticket at a time and has to go back to the end of the line (which happens instantaneously) in order to buy more tickets. If a person does not have any tickets left to buy, the person will leave the line. Return the time taken for the person at position k (0-indexed) to finish buying tickets.   Example 1: Input: tickets = [2,3,2], k = 2 Output: 6 Explanation: - In the first pass, everyone in the line buys a ticket and the line becomes [1, 2, 1]. - In the second pass, everyone in the line buys a ticket and the line becomes [0, 1, 0]. The person at position 2 has successfully bought 2 tickets and it took 3 + 3 = 6 seconds.

Example 2: Input: tickets = [5,1,1,1], k = 0 Output: 8 Explanation: - In the first pass, everyone in the line buys a ticket and the line becomes [4, 0, 0, 0]. -

In the next 4 passes, only the person in position 0 is buying tickets. The person at position 0 has successfully bought 5 tickets and it took 4 + 1 + 1 + 1 + 1 = 8 seconds.

Constraints:

n == tickets.length 1 <= n <= 100 1 <= tickets[i] <= 100 0 <= k < n

==
Problem Number: 390 URL: https://leetcode.com/problems/two-furthest-houses-with-different-colors Title: 2078. Two Furthest Houses With Different Colors Problem Description: There are n houses evenly lined up on the street, and each house is beautifully painted. You are given a 0-indexed integer array colors of length n, where colors[i] represents the color of the ith house. Return the maximum distance between two houses with different colors. The distance between the ith and jth houses is abs(i - j), where abs(x) is the absolute value of x. Example 1:

Input: colors = [1,1,1,6,1,1,1] Output: 3 Explanation: In the above image, color 1 is blue, and color 6 is red. The furthest two houses with different colors are house 0 and house 3. House 0 has color 1, and house 3 has color 6. The distance between them is abs(0 - 3) = 3. Note that houses 3 and 6 can also produce the optimal answer.

Example 2:

Input: colors = [1,8,3,8,3] Output: 4 Explanation: In the above image, color 1 is blue, color 8 is yellow, and color 3 is green. The furthest two houses with different colors are house 0 and house 4. House 0 has color 1, and house 4 has color 3. The distance between them is abs(0 - 4) = 4.

Example 3: Input: colors = [0,1] Output: 1 Explanation: The furthest two houses with different colors are house 0 and house 1. House 0 has color 0, and house 1 has color 1. The distance between them is abs(0 - 1) = 1.

Constraints:

n == colors.length 2 <= n <= 100 0 <= colors[i] <= 100 Test data are generated such that at least two houses have different colors.

==
Problem Number: 391 URL: https://leetcode.com/problems/count-common-words-with-one-occurrence Title: 2085. Count Common Words With One Occurrence Problem Description: Given two string arrays words1 and words2, return the number of strings that appear exactly once in each of the two arrays. Example 1: Input: words1 = ["leetcode","is","amazing","as","is"], words2 = ["amazing","leetcode","is"] Output: 2 Explanation: - "leetcode" appears exactly once in each of the two arrays. We count this string. - "amazing" appears exactly once in each of the two arrays. We count this string. - "is" appears in each of the two arrays, but there are 2 occurrences of it in words1. We do not count this string. - "as" appears once in words1, but does not

appear in words2. We do not count this string. Thus, there are 2 strings that appear exactly once in each of the two arrays.

Example 2: Input: words1 = ["b","bb","bbb"], words2 = ["a","aa","aaa"] Output: 0 Explanation: There are no strings that appear in each of the two arrays.

Example 3: Input: words1 = ["a","ab"], words2 = ["a","a","a","ab"] Output: 1 Explanation: The only string that appears exactly once in each of the two arrays is "ab".

Constraints:

1 <= words1.length, words2.length <= 1000 1 <= words1[i].length, words2[j].length <= 30 words1[i] and words2[j] consists only of lowercase English letters.

==
Problem Number: 392 URL: https://leetcode.com/problems/find-target-indices-after-sorting-array Title: 2089. Find Target Indices After Sorting Array Problem Description: You are given a 0-indexed integer array nums and a target element target. A target index is an index i such that nums[i] == target. Return a list of the target indices of nums after sorting nums in non-decreasing order. If there are no target indices, return an empty list. The returned list must be sorted in increasing order. Example 1: Input: nums = [1,2,5,2,3], target = 2 Output: [1,2] Explanation: After sorting, nums is [1,2,2,3,5]. The indices where nums[i] == 2 are 1 and 2.

Example 2: Input: nums = [1,2,5,2,3], target = 3 Output: [3] Explanation: After sorting, nums is [1,2,2,3,5]. The index where nums[i] == 3 is 3.

Example 3: Input: nums = [1,2,5,2,3], target = 5 Output: [4] Explanation: After sorting, nums is [1,2,2,3,5]. The index where nums[i] == 5 is 4.

Constraints:

1 <= nums.length <= 100 1 <= nums[i], target <= 100

==
Problem Number: 393 URL: https://leetcode.com/problems/finding-3-digit-even-numbers Title: 2094. Finding 3-Digit Even Numbers Problem Description: You are given an integer array digits, where each element is a digit. The array may contain duplicates. You need to find all the unique integers that follow the given requirements:

The integer consists of the concatenation of three elements from digits in any arbitrary order. The integer does not have leading zeros. The integer is even.

For example, if the given digits were [1, 2, 3], integers 132 and 312 follow the requirements. Return a sorted array of the unique integers. Example 1: Input: digits = [2,1,3,0] Output: [102,120,130,132,210,230,302,310,312,320] Explanation: All the possible integers that follow the requirements are in the output array. Notice that there are no odd integers or integers with leading zeros.

Example 2: Input: digits = [2,2,8,8,2] Output: [222,228,282,288,822,828,882] Explanation: The same digit can be used as many times as it appears in digits. In this example, the digit 8 is used twice each time in 288, 828, and 882.

Example 3: Input: digits = [3,7,5] Output: [] Explanation: No even integers can be formed using the given digits.

Constraints:

3 <= digits.length <= 100 0 <= digits[i] <= 9

==
Problem Number: 394 URL: https://leetcode.com/problems/find-subsequence-of-length-k-with-the-largest-sum Title: 2099. Find Subsequence of Length K With the Largest Sum Problem Description: You are given an integer array nums and an integer k. You want to find a subsequence of nums of length k that has the largest sum. Return any such subsequence as an integer array of length k. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: nums = [2,1,3,3], k = 2 Output: [3,3] Explanation: The subsequence has the largest sum of 3 + 3 = 6. Example 2: Input: nums = [-1,-2,3,4], k = 3 Output: [-1,3,4] Explanation: The subsequence has the largest sum of -1 + 3 + 4 = 6.

Example 3: Input: nums = [3,4,3,3], k = 2 Output: [3,4] Explanation: The subsequence has the largest sum of 3 + 4 = 7. Another possible subsequence is [4, 3].

Constraints:

1 <= nums.length <= 1000 -105 <= nums[i] <= 105 1 <= k <= nums.length

==
Problem Number: 395 URL: https://leetcode.com/problems/rings-and-rods Title: 2103. Rings and Rods Problem Description: There are n rings and each ring is either red, green, or blue. The rings are distributed across ten rods labeled from 0 to 9. You are given a string rings of length 2n that describes the n rings that are placed onto the rods. Every two characters in rings forms a color-position pair that is used to describe each ring where:

The first character of the ith pair denotes the ith ring's color ('R', 'G', 'B'). The second character of the ith pair denotes the rod that the ith ring is placed on ('0' to '9').

For example, "R3G2B1" describes n == 3 rings: a red ring placed onto the rod labeled 3, a green ring placed onto the rod labeled 2, and a blue ring placed onto the rod labeled 1. Return the number of rods that have all three colors of rings on them. Example 1:

Input: rings = ”B0B6G0R6R0R6G9” Output: 1 Explanation: - The rod labeled 0 holds 3 rings with all colors: red, green, and blue. - The rod labeled 6 holds 3 rings, but it only has red and blue. - The rod labeled 9 holds only a green ring. Thus, the number of rods with all three colors is 1.

Example 2:

Input: rings = ”B0R0G0R9R0B0G0” Output: 1 Explanation: - The rod labeled 0 holds 6 rings with all colors: red, green, and blue. - The rod labeled 9 holds only a red ring. Thus, the number of rods with all three colors is 1.

Example 3: Input: rings = ”G4” Output: 0 Explanation: Only one ring is given. Thus, no rods have all three colors.

Constraints:

rings.length == 2 * n 1 <= n <= 100 rings[i] where i is even is either ’R’, ’G’, or ’B’ (0-indexed). rings[i] where i is odd is a digit from ’0’ to ’9’ (0-indexed).

==
Problem Number: 396 URL: https://leetcode.com/problems/find-first-palindromic-string-in-the-array Title: 2108. Find First Palindromic String in the Array Problem Description: Given an array of strings words, return the first palindromic string in the array. If there is no such string, return an empty string ””. A string is palindromic if it reads the same forward and backward. Example 1: Input: words = [”abc”,”car”,”ada”,”racecar”,”cool”] Output: ”ada” Explanation: The first string that is palindromic is ”ada”. Note that ”racecar” is also palindromic, but it is not the first.

Example 2: Input: words = [”notapalindrome”,”racecar”] Output: ”racecar” Explanation: The first and only string that is palindromic is ”racecar”.

Example 3: Input: words = [”def”,”ghi”] Output: ”” Explanation: There are no palindromic strings, so the empty string is returned.

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 100 words[i] consists only of lowercase English letters.

==
Problem Number: 397 URL: https://leetcode.com/problems/maximum-number-of-words-found-in-sentences Title: 2114. Maximum Number of Words Found in Sentences Problem Description: A sentence is a list of words that are separated by a single space with no leading or trailing spaces. You are given an array of strings sentences, where each sentences[i] represents a single sentence. Return the maximum number of words that appear in a single sentence. Example 1: Input: sentences = [”alice and bob love leetcode”, ”i think so too”, ”this is great thanks very much”] Output: 6 Explanation: - The first sentence, ”alice and bob love leetcode”, has 5 words in total. - The second sentence, ”i think so too”, has 4 words in total. - The third sentence, ”this is

great thanks very much", has 6 words in total. Thus, the maximum number of words in a single sentence comes from the third sentence, which has 6 words.

Example 2: Input: sentences = ["please wait", "continue to fight", "continue to win"] Output: 3 Explanation: It is possible that multiple sentences contain the same number of words. In this example, the second and third sentences (underlined) have the same number of words.

Constraints:

1 <= sentences.length <= 100 1 <= sentences[i].length <= 100 sentences[i] consists only of lowercase English letters and ' ' only. sentences[i] does not have leading or trailing spaces. All the words in sentences[i] are separated by a single space.

==
Problem Number: 398 URL: https://leetcode.com/problems/a-number-after-a-double-reversal Title: 2119. A Number After a Double Reversal Problem Description: Reversing an integer means to reverse all its digits.

For example, reversing 2021 gives 1202. Reversing 12300 gives 321 as the leading zeros are not retained.

Given an integer num, reverse num to get reversed1, then reverse reversed1 to get reversed2. Return true if reversed2 equals num. Otherwise return false. Example 1: Input: num = 526 Output: true Explanation: Reverse num to get 625, then reverse 625 to get 526, which equals num.

Example 2: Input: num = 1800 Output: false Explanation: Reverse num to get 81, then reverse 81 to get 18, which does not equal num.

Example 3: Input: num = 0 Output: true Explanation: Reverse num to get 0, then reverse 0 to get 0, which equals num.

Constraints:

0 <= num <= 106

==
Problem Number: 399 URL: https://leetcode.com/problems/check-if-all-as-appears-before-all-bs Title: 2124. Check if All A's Appears Before All B's Problem Description: Given a string s consisting of only the characters 'a' and 'b', return true if every 'a' appears before every 'b' in the string. Otherwise, return false. Example 1: Input: s = "aaabbb" Output: true Explanation: The 'a's are at indices 0, 1, and 2, while the 'b's are at indices 3, 4, and 5. Hence, every 'a' appears before every 'b' and we return true.

Example 2: Input: s = "abab" Output: false Explanation: There is an 'a' at index 2 and a 'b' at index 1. Hence, not every 'a' appears before every 'b' and we return false.

Example 3: Input: s = ”bbb” Output: true Explanation: There are no ’a’s, hence, every ’a’ appears before every ’b’ and we return true.

Constraints:

1 <= s.length <= 100 s[i] is either ’a’ or ’b’.

==

Problem Number: 400 URL: https://leetcode.com/problems/capitalize-the-title Title: 2129. Capitalize the Title Problem Description: You are given a string title consisting of one or more words separated by a single space, where each word consists of English letters. Capitalize the string by changing the capitalization of each word such that:

If the length of the word is 1 or 2 letters, change all letters to lowercase. Otherwise, change the first letter to uppercase and the remaining letters to lowercase.

Return the capitalized title. Example 1: Input: title = ”capiTalIze tHe titLe” Output: ”Capitalize The Title” Explanation: Since all the words have a length of at least 3, the first letter of each word is uppercase, and the remaining letters are lowercase.

Example 2: Input: title = ”First leTTeR of EACH Word” Output: ”First Letter of Each Word” Explanation: The word ”of” has length 2, so it is all lowercase. The remaining words have a length of at least 3, so the first letter of each remaining word is uppercase, and the remaining letters are lowercase.

Example 3: Input: title = ”i lOve leetcode” Output: ”i Love Leetcode” Explanation: The word ”i” has length 1, so it is lowercase. The remaining words have a length of at least 3, so the first letter of each remaining word is uppercase, and the remaining letters are lowercase.

Constraints:

1 <= title.length <= 100 title consists of words separated by a single space without any leading or trailing spaces. Each word consists of uppercase and lowercase English letters and is non-empty.

==

Problem Number: 401 URL: https://leetcode.com/problems/check-if-every-row-and-column-contains-all-numbers Title: 2133. Check if Every Row and Column Contains All Numbers Problem Description: An n x n matrix is valid if every row and every column contains all the integers from 1 to n (inclusive). Given an n x n integer matrix matrix, return true if the matrix is valid. Otherwise, return false. Example 1:

Input: matrix = [[1,2,3],[3,1,2],[2,3,1]] Output: true Explanation: In this case, n = 3, and every row and column contains the numbers 1, 2, and 3. Hence, we return true.

Example 2:

Input: matrix = [[1,1,1],[1,2,3],[1,2,3]] Output: false Explanation: In this case, n = 3, but the first row and the first column do not contain the numbers 2 or 3. Hence, we return false.

Constraints:

n == matrix.length == matrix[i].length 1 <= n <= 100 1 <= matrix[i][j] <= n

==
Problem Number: 402 URL: https://leetcode.com/problems/divide-a-string-into-groups-of-size-k Title: 2138. Divide a String Into Groups of Size k Problem Description: A string s can be partitioned into groups of size k using the following procedure:

The first group consists of the first k characters of the string, the second group consists of the next k characters of the string, and so on. Each character can be a part of exactly one group. For the last group, if the string does not have k characters remaining, a character fill is used to complete the group.

Note that the partition is done so that after removing the fill character from the last group (if it exists) and concatenating all the groups in order, the resultant string should be s. Given the string s, the size of each group k and the character fill, return a string array denoting the composition of every group s has been divided into, using the above procedure. Example 1: Input: s = "abcdefghi", k = 3, fill = "x" Output: ["abc","def","ghi"] Explanation: The first 3 characters "abc" form the first group. The next 3 characters "def" form the second group. The last 3 characters "ghi" form the third group. Since all groups can be completely filled by characters from the string, we do not need to use fill. Thus, the groups formed are "abc", "def", and "ghi".

Example 2: Input: s = "abcdefghij", k = 3, fill = "x" Output: ["abc","def","ghi","jxx"] Explanation: Similar to the previous example, we are forming the first three groups "abc", "def", and "ghi". For the last group, we can only use the character 'j' from the string. To complete this group, we add 'x' twice. Thus, the 4 groups formed are "abc", "def", "ghi", and "jxx".

Constraints:

1 <= s.length <= 100 s consists of lowercase English letters only. 1 <= k <= 100 fill is a lowercase English letter.

==
Problem Number: 403 URL: https://leetcode.com/problems/minimum-cost-of-buying-candies-with-discount Title: 2144. Minimum Cost of Buying Candies With Discount Problem Description: A shop is selling candies at a discount. For every two candies sold, the shop gives a third candy for free. The customer can choose any candy to take away for free as long as the cost of the chosen candy is less than or equal to the minimum cost of the two candies bought.

For example, if there are 4 candies with costs 1, 2, 3, and 4, and the customer buys candies with costs 2 and 3, they can take the candy with cost 1 for free, but not the candy with cost 4.

Given a 0-indexed integer array cost, where cost[i] denotes the cost of the ith candy, return the minimum cost of buying all the candies. Example 1: Input: cost = [1,2,3] Output: 5 Explanation: We buy the candies with costs 2 and 3, and take the candy with cost 1 for free. The total cost of buying all candies is 2 + 3 = 5. This is the only way we can buy the candies. Note that we cannot buy candies with costs 1 and 3, and then take the candy with cost 2 for free. The cost of the free candy has to be less than or equal to the minimum cost of the purchased candies.

Example 2: Input: cost = [6,5,7,9,2,2] Output: 23 Explanation: The way in which we can get the minimum cost is described below: - Buy candies with costs 9 and 7 - Take the candy with cost 6 for free - We buy candies with costs 5 and 2 - Take the last remaining candy with cost 2 for free Hence, the minimum cost to buy all candies is 9 + 7 + 5 + 2 = 23.

Example 3: Input: cost = [5,5] Output: 10 Explanation: Since there are only 2 candies, we buy both of them. There is not a third candy we can take for free. Hence, the minimum cost to buy all candies is 5 + 5 = 10.

Constraints:

1 <= cost.length <= 100 1 <= cost[i] <= 100

==

Problem Number: 404 URL: https://leetcode.com/problems/count-elements-with-strictly-smaller-and-greater-elements Title: 2148. Count Elements With Strictly Smaller and Greater Elements Problem Description: Given an integer array nums, return the number of elements that have both a strictly smaller and a strictly greater element appear in nums. Example 1: Input: nums = [11,7,2,15] Output: 2 Explanation: The element 7 has the element 2 strictly smaller than it and the element 11 strictly greater than it. Element 11 has element 7 strictly smaller than it and element 15 strictly greater than it. In total there are 2 elements having both a strictly smaller and a strictly greater element appear in nums.

Example 2: Input: nums = [-3,3,3,90] Output: 2 Explanation: The element 3 has the element -3 strictly smaller than it and the element 90 strictly greater than it. Since there are two elements with the value 3, in total there are 2 elements having both a strictly smaller and a strictly greater element appear in nums.

Constraints:

1 <= nums.length <= 100 -105 <= nums[i] <= 105

==

Problem Number: 405 URL: https://leetcode.com/problems/keep-multiplying-found-values-by-two Title: 2154. Keep Multiplying Found Values by Two Problem Description: You are given an array of integers nums. You are also given an integer original which is the first number that needs to be searched for in nums. You then do the following steps:

If original is found in nums, multiply it by two (i.e., set original = 2 * original). Otherwise, stop the process. Repeat this process with the new number as long as you keep finding the number.

Return the final value of original. Example 1: Input: nums = [5,3,6,1,12], original = 3 Output: 24 Explanation: - 3 is found in nums. 3 is multiplied by 2 to obtain 6. - 6 is found in nums. 6 is multiplied by 2 to obtain 12. - 12 is found in nums. 12 is multiplied by 2 to obtain 24. - 24 is not found in nums. Thus, 24 is returned.

Example 2: Input: nums = [2,7,9], original = 4 Output: 4 Explanation: - 4 is not found in nums. Thus, 4 is returned.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i], original <= 1000

==

Problem Number: 406 URL: https://leetcode.com/problems/minimum-sum-of-four-digit-number-after-splitting-digits Title: 2160. Minimum Sum of Four Digit Number After Splitting Digits Problem Description: You are given a positive integer num consisting of exactly four digits. Split num into two new integers new1 and new2 by using the digits found in num. Leading zeros are allowed in new1 and new2, and all the digits found in num must be used.

For example, given num = 2932, you have the following digits: two 2's, one 9 and one 3. Some of the possible pairs [new1, new2] are [22, 93], [23, 92], [223, 9] and [2, 329].

Return the minimum possible sum of new1 and new2. Example 1: Input: num = 2932 Output: 52 Explanation: Some possible pairs [new1, new2] are [29, 23], [223, 9], etc. The minimum sum can be obtained by the pair [29, 23]: 29 + 23 = 52.

Example 2: Input: num = 4009 Output: 13 Explanation: Some possible pairs [new1, new2] are [0, 49], [490, 0], etc. The minimum sum can be obtained by the pair [4, 9]: 4 + 9 = 13.

Constraints:

1000 <= num <= 9999

==

Problem Number: 407 URL: https://leetcode.com/problems/sort-even-and-odd-indices-independently Title: 2164. Sort Even and Odd Indices Independently Problem Description: You are given a 0-indexed integer array nums. Rearrange the values of nums according to the following rules:

Sort the values at odd indices of nums in non-increasing order.

For example, if nums = [4,1,2,3] before this step, it becomes [4,3,2,1] after. The values at odd indices 1 and 3 are sorted in non-increasing order.

Sort the values at even indices of nums in non-decreasing order.

For example, if nums = [4,1,2,3] before this step, it becomes [2,1,4,3] after. The values at even indices 0 and 2 are sorted in non-decreasing order.

Return the array formed after rearranging the values of nums.   Example 1: Input: nums = [4,1,2,3] Output: [2,3,4,1] Explanation: First, we sort the values present at odd indices (1 and 3) in non-increasing order. So, nums changes from [4,1,2,3] to [4,3,2,1]. Next, we sort the values present at even indices (0 and 2) in non-decreasing order. So, nums changes from [4,1,2,3] to [2,3,4,1]. Thus, the array formed after rearranging the values is [2,3,4,1].

Example 2: Input: nums = [2,1] Output: [2,1] Explanation: Since there is exactly one odd index and one even index, no rearrangement of values takes place. The resultant array formed is [2,1], which is the same as the initial array.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100

==
Problem Number: 408 URL: https://leetcode.com/problems/count-operations-to-obtain-zero Title: 2169. Count Operations to Obtain Zero Problem Description: You are given two non-negative integers num1 and num2. In one operation, if num1 >= num2, you must subtract num2 from num1, otherwise subtract num1 from num2.

For example, if num1 = 5 and num2 = 4, subtract num2 from num1, thus obtaining num1 = 1 and num2 = 4. However, if num1 = 4 and num2 = 5, after one operation, num1 = 4 and num2 = 1.

Return the number of operations required to make either num1 = 0 or num2 = 0.   Example 1: Input: num1 = 2, num2 = 3 Output: 3 Explanation: - Operation 1: num1 = 2, num2 = 3. Since num1 < num2, we subtract num1 from num2 and get num1 = 2, num2 = 3 - 2 = 1. - Operation 2: num1 = 2, num2 = 1. Since num1 > num2, we subtract num2 from num1. - Operation 3: num1 = 1, num2 = 1. Since num1 == num2, we subtract num2 from num1. Now num1 = 0 and num2 = 1. Since num1 == 0, we do not need to perform any further operations. So the total number of operations required is 3.

Example 2: Input: num1 = 10, num2 = 10 Output: 1 Explanation: - Operation 1: num1 = 10, num2 = 10. Since num1 == num2, we subtract num2 from num1 and get num1 = 10 - 10 = 0. Now num1 = 0 and num2 = 10. Since num1 == 0, we are done. So the total number of operations required is 1.

Constraints:

0 <= num1, num2 <= 105

==

Problem Number: 409 URL: https://leetcode.com/problems/count-equal-and-divisible-pairs-in-an-array Title: 2176. Count Equal and Divisible Pairs in an Array Problem Description: Given a 0-indexed integer array nums of length n and an integer k, return the number of pairs (i, j) where 0 <= i < j < n, such that nums[i] == nums[j] and (i * j) is divisible by k. Example 1: Input: nums = [3,1,2,2,2,1,3], k = 2 Output: 4 Explanation: There are 4 pairs that meet all the requirements: - nums[0] == nums[6], and 0 * 6 == 0, which is divisible by 2. - nums[2] == nums[3], and 2 * 3 == 6, which is divisible by 2. - nums[2] == nums[4], and 2 * 4 == 8, which is divisible by 2. - nums[3] == nums[4], and 3 * 4 == 12, which is divisible by 2.

Example 2: Input: nums = [1,2,3,4], k = 1 Output: 0 Explanation: Since no value in nums is repeated, there are no pairs (i,j) that meet all the requirements.

Constraints:

1 <= nums.length <= 100 1 <= nums[i], k <= 100

==

Problem Number: 410 URL: https://leetcode.com/problems/count-integers-with-even-digit-sum Title: 2180. Count Integers With Even Digit Sum Problem Description: Given a positive integer num, return the number of positive integers less than or equal to num whose digit sums are even. The digit sum of a positive integer is the sum of all its digits. Example 1: Input: num = 4 Output: 2 Explanation: The only integers less than or equal to 4 whose digit sums are even are 2 and 4.

Example 2: Input: num = 30 Output: 14 Explanation: The 14 integers less than or equal to 30 whose digit sums are even are 2, 4, 6, 8, 11, 13, 15, 17, 19, 20, 22, 24, 26, and 28.

Constraints:

1 <= num <= 1000

==

Problem Number: 411 URL: https://leetcode.com/problems/counting-words-with-a-given-prefix Title: 2185. Counting Words With a Given Prefix Problem Description: You are given an array of strings words and a string pref. Return the number of strings in words that contain pref as a prefix. A prefix of a string s is any leading contiguous substring of s. Example 1: Input: words = [”pay”,”attention”,”practice”,”attend”], pref = ”at” Output: 2 Explanation: The 2 strings that contain ”at” as a prefix are: ”attention” and ”attend”.

Example 2: Input: words = [”leetcode”,”win”,”loops”,”success”], pref = ”code” Output: 0 Explanation: There are no strings that contain ”code” as a prefix.

Constraints:

1 <= words.length <= 100 1 <= words[i].length, pref.length <= 100 words[i] and pref consist of lowercase English letters.

==
Problem Number: 412 URL: https://leetcode.com/problems/most-frequent-number-following-key-in-an-array Title: 2190. Most Frequent Number Following Key In an Array Problem Description: You are given a 0-indexed integer array nums. You are also given an integer key, which is present in nums. For every unique integer target in nums, count the number of times target immediately follows an occurrence of key in nums. In other words, count the number of indices i such that:

0 <= i <= nums.length - 2, nums[i] == key and, nums[i + 1] == target.

Return the target with the maximum count. The test cases will be generated such that the target with maximum count is unique. Example 1: Input: nums = [1,100,200,1,100], key = 1 Output: 100 Explanation: For target = 100, there are 2 occurrences at indices 1 and 4 which follow an occurrence of key. No other integers follow an occurrence of key, so we return 100.

Example 2: Input: nums = [2,2,2,2,3], key = 2 Output: 2 Explanation: For target = 2, there are 3 occurrences at indices 1, 2, and 3 which follow an occurrence of key. For target = 3, there is only one occurrence at index 4 which follows an occurrence of key. target = 2 has the maximum number of occurrences following an occurrence of key, so we return 2.

Constraints:

2 <= nums.length <= 1000 1 <= nums[i] <= 1000 The test cases will be generated such that the answer is unique.

==
Problem Number: 413 URL: https://leetcode.com/problems/cells-in-a-range-on-an-excel-sheet Title: 2194. Cells in a Range on an Excel Sheet Problem Description: A cell (r, c) of an excel sheet is represented as a string "<col><row>" where:

<col> denotes the column number c of the cell. It is represented by alphabetical letters.

For example, the 1st column is denoted by 'A', the 2nd by 'B', the 3rd by 'C', and so on.

<row> is the row number r of the cell. The rth row is represented by the integer r.

You are given a string s in the format "<col1><row1>:<col2><row2>", where <col1> represents the column c1, <row1> represents the row r1, <col2> represents the column c2, and <row2> represents the row r2, such that r1 <= r2 and c1 <= c2. Return the list of cells (x, y) such that r1 <= x <= r2 and c1 <= y <= c2. The cells should be represented as strings in the format mentioned above and be sorted in non-decreasing order first by columns and then by rows. Example 1:

Input: s = "K1:L2" Output: ["K1","K2","L1","L2"] Explanation: The above diagram shows the cells which should be present in the list. The red arrows denote the order in which the cells should be presented.

Example 2:

Input: s = "A1:F1" Output: ["A1","B1","C1","D1","E1","F1"] Explanation: The above diagram shows the cells which should be present in the list. The red arrow denotes the order in which the cells should be presented.

Constraints:

s.length == 5 'A' <= s[0] <= s[3] <= 'Z' '1' <= s[1] <= s[4] <= '9' s consists of uppercase English letters, digits and ':'.

==

Problem Number: 414 URL: https://leetcode.com/problems/find-all-k-distant-indices-in-an-array Title: 2200. Find All K-Distant Indices in an Array Problem Description: You are given a 0-indexed integer array nums and two integers key and k. A k-distant index is an index i of nums for which there exists at least one index j such that |i - j| <= k and nums[j] == key. Return a list of all k-distant indices sorted in increasing order. Example 1: Input: nums = [3,4,9,1,3,9,5], key = 9, k = 1 Output: [1,2,3,4,5,6] Explanation: Here, nums[2] == key and nums[5] == key. - For index 0, |0 - 2| > k and |0 - 5| > k, so there is no j where |0 - j| <= k and nums[j] == key. Thus, 0 is not a k-distant index. - For index 1, |1 - 2| <= k and nums[2] == key, so 1 is a k-distant index. - For index 2, |2 - 2| <= k and nums[2] == key, so 2 is a k-distant index. - For index 3, |3 - 2| <= k and nums[2] == key, so 3 is a k-distant index. - For index 4, |4 - 5| <= k and nums[5] == key, so 4 is a k-distant index. - For index 5, |5 - 5| <= k and nums[5] == key, so 5 is a k-distant index. - For index 6, |6 - 5| <= k and nums[5] == key, so 6 is a k-distant index. Thus, we return [1,2,3,4,5,6] which is sorted in increasing order.

Example 2: Input: nums = [2,2,2,2,2], key = 2, k = 2 Output: [0,1,2,3,4] Explanation: For all indices i in nums, there exists some index j such that |i - j| <= k and nums[j] == key, so every index is a k-distant index. Hence, we return [0,1,2,3,4].

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 1000 key is an integer from the array nums. 1 <= k <= nums.length

==

Problem Number: 415 URL: https://leetcode.com/problems/divide-array-into-equal-pairs Title: 2206. Divide Array Into Equal Pairs Problem Description: You are given an integer array nums consisting of 2 * n integers. You need to divide nums into n pairs such that:

Each element belongs to exactly one pair. The elements present in a pair are equal.

Return true if nums can be divided into n pairs, otherwise return false. Example 1: Input: nums = [3,2,3,2,2,2] Output: true Explanation: There are 6 elements in nums, so they should be divided into 6 / 2 = 3 pairs. If nums is divided into the pairs (2, 2), (3, 3), and (2, 2), it will satisfy all the conditions.

Example 2: Input: nums = [1,2,3,4] Output: false Explanation: There is no way to divide nums into 4 / 2 = 2 pairs such that the pairs satisfy every condition.

Constraints:

nums.length == 2 * n 1 <= n <= 500 1 <= nums[i] <= 500

==
Problem Number: 416 URL: https://leetcode.com/problems/count-hills-and-valleys-in-an-array Title: 2210. Count Hills and Valleys in an Array Problem Description: You are given a 0-indexed integer array nums. An index i is part of a hill in nums if the closest non-equal neighbors of i are smaller than nums[i]. Similarly, an index i is part of a valley in nums if the closest non-equal neighbors of i are larger than nums[i]. Adjacent indices i and j are part of the same hill or valley if nums[i] == nums[j]. Note that for an index to be part of a hill or valley, it must have a non-equal neighbor on both the left and right of the index. Return the number of hills and valleys in nums. Example 1: Input: nums = [2,4,1,1,6,5] Output: 3 Explanation: At index 0: There is no non-equal neighbor of 2 on the left, so index 0 is neither a hill nor a valley. At index 1: The closest non-equal neighbors of 4 are 2 and 1. Since 4 > 2 and 4 > 1, index 1 is a hill. At index 2: The closest non-equal neighbors of 1 are 4 and 6. Since 1 < 4 and 1 < 6, index 2 is a valley. At index 3: The closest non-equal neighbors of 1 are 4 and 6. Since 1 < 4 and 1 < 6, index 3 is a valley, but note that it is part of the same valley as index 2. At index 4: The closest non-equal neighbors of 6 are 1 and 5. Since 6 > 1 and 6 > 5, index 4 is a hill. At index 5: There is no non-equal neighbor of 5 on the right, so index 5 is neither a hill nor a valley. There are 3 hills and valleys so we return 3.

Example 2: Input: nums = [6,6,5,5,4,1] Output: 0 Explanation: At index 0: There is no non-equal neighbor of 6 on the left, so index 0 is neither a hill nor a valley. At index 1: There is no non-equal neighbor of 6 on the left, so index 1 is neither a hill nor a valley. At index 2: The closest non-equal neighbors of 5 are 6 and 4. Since 5 < 6 and 5 > 4, index 2 is neither a hill nor a valley. At index 3: The closest non-equal neighbors of 5 are 6 and 4. Since 5 < 6 and 5 > 4, index 3 is neither a hill nor a valley. At index 4: The closest non-equal neighbors of 4 are 5 and 1. Since 4 < 5 and 4 > 1, index 4 is neither a hill nor a valley. At index 5: There is no non-equal neighbor of 1 on the right, so index 5 is neither a hill nor a valley. There are 0 hills and valleys so we return 0.

Constraints:

3 <= nums.length <= 100 1 <= nums[i] <= 100

==
Problem Number: 417 URL: https://leetcode.com/problems/find-the-difference-of-two-arrays Title: 2215. Find the Difference of Two Arrays Problem Description: Given two 0-indexed integer arrays nums1 and nums2, return a list answer of size 2 where:

answer[0] is a list of all distinct integers in nums1 which are not present in nums2. answer[1] is a list of all distinct integers in nums2 which are not present in nums1.

Note that the integers in the lists may be returned in any order. Example 1: Input: nums1 = [1,2,3], nums2 = [2,4,6] Output: [[1,3],[4,6]] Explanation: For nums1, nums1[1] = 2 is present at index 0 of nums2, whereas nums1[0] = 1 and nums1[2] = 3 are not present in nums2. Therefore, answer[0] = [1,3]. For nums2, nums2[0] = 2 is present at index 1 of nums1, whereas nums2[1] = 4 and nums2[2] = 6 are not present in nums2. Therefore, answer[1] = [4,6]. Example 2: Input: nums1 = [1,2,3,3], nums2 = [1,1,2,2] Output: [[3],[]] Explanation: For nums1, nums1[2] and nums1[3] are not present in nums2. Since nums1[2] == nums1[3], their value is only included once and answer[0] = [3]. Every integer in nums2 is present in nums1. Therefore, answer[1] = [].

Constraints:

1 <= nums1.length, nums2.length <= 1000 -1000 <= nums1[i], nums2[i] <= 1000

==
Problem Number: 418 URL: https://leetcode.com/problems/minimum-bit-flips-to-convert-number Title: 2220. Minimum Bit Flips to Convert Number Problem Description: A bit flip of a number x is choosing a bit in the binary representation of x and flipping it from either 0 to 1 or 1 to 0.

For example, for x = 7, the binary representation is 111 and we may choose any bit (including any leading zeros not shown) and flip it. We can flip the first bit from the right to get 110, flip the second bit from the right to get 101, flip the fifth bit from the right (a leading zero) to get 10111, etc.

Given two integers start and goal, return the minimum number of bit flips to convert start to goal. Example 1: Input: start = 10, goal = 7 Output: 3 Explanation: The binary representation of 10 and 7 are 1010 and 0111 respectively. We can convert 10 to 7 in 3 steps: - Flip the first bit from the right: 1010 -> 1011. - Flip the third bit from the right: 1011 -> 1111. - Flip the fourth bit from the right: 1111 -> 0111. It can be shown we cannot convert 10 to 7 in less than 3 steps. Hence, we return 3. Example 2: Input: start = 3, goal = 4 Output: 3 Explanation: The binary representation of 3 and 4 are 011 and 100 respectively. We can convert 3 to 4 in 3 steps: - Flip the first bit from the right: 011 -> 010. - Flip the second bit from the right: 010 -> 000. - Flip the third bit from the right: 000 -> 100. It can be shown we cannot convert 3 to 4 in less than 3 steps. Hence, we return 3.

Constraints:

0 <= start, goal <= 109

===

Problem Number: 419 URL: https://leetcode.com/problems/minimum-number-of-operations-to-convert-time Title: 2224. Minimum Number of Operations to Convert Time Problem Description: You are given two strings current and correct representing two 24-hour times. 24-hour times are formatted as ”HH:MM”, where HH is between 00 and 23, and MM is between 00 and 59. The earliest 24-hour time is 00:00, and the latest is 23:59. In one operation you can increase the time current by 1, 5, 15, or 60 minutes. You can perform this operation any number of times. Return the minimum number of operations needed to convert current to correct. Example 1: Input: current = ”02:30”, correct = ”04:35” Output: 3 Explanation: We can convert current to correct in 3 operations as follows: - Add 60 minutes to current. current becomes ”03:30”. - Add 60 minutes to current. current becomes ”04:30”. - Add 5 minutes to current. current becomes ”04:35”. It can be proven that it is not possible to convert current to correct in fewer than 3 operations. Example 2: Input: current = ”11:00”, correct = ”11:01” Output: 1 Explanation: We only have to add one minute to current, so the minimum number of operations needed is 1.

Constraints:

current and correct are in the format ”HH:MM” current <= correct

===

Problem Number: 420 URL: https://leetcode.com/problems/largest-number-after-digit-swaps-by-parity Title: 2231. Largest Number After Digit Swaps by Parity Problem Description: You are given a positive integer num. You may swap any two digits of num that have the same parity (i.e. both odd digits or both even digits). Return the largest possible value of num after any number of swaps. Example 1: Input: num = 1234 Output: 3412 Explanation: Swap the digit 3 with the digit 1, this results in the number 3214. Swap the digit 2 with the digit 4, this results in the number 3412. Note that there may be other sequences of swaps but it can be shown that 3412 is the largest possible number. Also note that we may not swap the digit 4 with the digit 1 since they are of different parities.

Example 2: Input: num = 65875 Output: 87655 Explanation: Swap the digit 8 with the digit 6, this results in the number 85675. Swap the first digit 5 with the digit 7, this results in the number 87655. Note that there may be other sequences of swaps but it can be shown that 87655 is the largest possible number.

Constraints:

1 <= num <= 109

===

Problem Number: 421 URL: https://leetcode.com/problems/add-two-integers Title: 2235. Add Two Integers Problem Description: Given two integers num1 and num2, return the sum of the two integers. Example 1: Input: num1 = 12, num2 = 5 Output: 17 Explanation: num1 is 12, num2 is 5, and their sum is 12 + 5 = 17, so 17 is returned.

Example 2: Input: num1 = -10, num2 = 4 Output: -6 Explanation: num1 + num2 = -6, so -6 is returned.

Constraints:

-100 <= num1, num2 <= 100

==

Problem Number: 422 URL: https://leetcode.com/problems/root-equals-sum-of-children Title: 2236. Root Equals Sum of Children Problem Description: You are given the root of a binary tree that consists of exactly 3 nodes: the root, its left child, and its right child. Return true if the value of the root is equal to the sum of the values of its two children, or false otherwise. Example 1:

Input: root = [10,4,6] Output: true Explanation: The values of the root, its left child, and its right child are 10, 4, and 6, respectively. 10 is equal to 4 + 6, so we return true.

Example 2:

Input: root = [5,3,1] Output: false Explanation: The values of the root, its left child, and its right child are 5, 3, and 1, respectively. 5 is not equal to 3 + 1, so we return false.

Constraints:

The tree consists only of the root, its left child, and its right child. -100 <= Node.val <= 100

==

Problem Number: 423 URL: https://leetcode.com/problems/find-closest-number-to-zero Title: 2239. Find Closest Number to Zero Problem Description: Given an integer array nums of size n, return the number with the value closest to 0 in nums. If there are multiple answers, return the number with the largest value. Example 1: Input: nums = [-4,-2,1,4,8] Output: 1 Explanation: The distance from -4 to 0 is |-4| = 4. The distance from -2 to 0 is |-2| = 2. The distance from 1 to 0 is |1| = 1. The distance from 4 to 0 is |4| = 4. The distance from 8 to 0 is |8| = 8. Thus, the closest number to 0 in the array is 1.

Example 2: Input: nums = [2,-1,1] Output: 1 Explanation: 1 and -1 are both the closest numbers to 0, so 1 being larger is returned.

Constraints:

1 <= n <= 1000 -105 <= nums[i] <= 105

== Problem Number: 424 URL: https://leetcode.com/problems/calculate-digit-sum-of-a-string Title: 2243. Calculate Digit Sum of a String Problem Description: You are given a string s consisting of digits and an integer k. A round can be completed if the length of s is greater than k. In one round, do the following:

Divide s into consecutive groups of size k such that the first k characters are in the first group, the next k characters are in the second group, and so on. Note that the size of the last group can be smaller than k. Replace each group of s with a string representing the sum of all its digits. For example, "346" is replaced with "13" because 3 + 4 + 6 = 13. Merge consecutive groups together to form a new string. If the length of the string is greater than k, repeat from step 1.

Return s after all rounds have been completed. Example 1: Input: s = "11111222223", k = 3 Output: "135" Explanation: - For the first round, we divide s into groups of size 3: "111", "112", "222", and "23". Then we calculate the digit sum of each group: 1 + 1 + 1 = 3, 1 + 1 + 2 = 4, 2 + 2 + 2 = 6, and 2 + 3 = 5. So, s becomes "3" + "4" + "6" + "5" = "3465" after the first round. - For the second round, we divide s into "346" and "5". Then we calculate the digit sum of each group: 3 + 4 + 6 = 13, 5 = 5. So, s becomes "13" + "5" = "135" after second round. Now, s.length <= k, so we return "135" as the answer.

Example 2: Input: s = "00000000", k = 3 Output: "000" Explanation: We divide s into "000", "000", and "00". Then we calculate the digit sum of each group: 0 + 0 + 0 = 0, 0 + 0 + 0 = 0, and 0 + 0 = 0. s becomes "0" + "0" + "0" = "000", whose length is equal to k, so we return "000".

Constraints:

1 <= s.length <= 100 2 <= k <= 100 s consists of digits only.

== Problem Number: 425 URL: https://leetcode.com/problems/intersection-of-multiple-arrays Title: 2248. Intersection of Multiple Arrays Problem Description: Given a 2D integer array nums where nums[i] is a non-empty array of distinct positive integers, return the list of integers that are present in each array of nums sorted in ascending order. Example 1: Input: nums = [[3,1,2,4,5],[1,2,3,4],[3,4,5,6]] Output: [3,4] Explanation: The only integers present in each of nums[0] = [3,1,2,4,5], nums[1] = [1,2,3,4], and nums[2] = [3,4,5,6] are 3 and 4, so we return [3,4]. Example 2: Input: nums = [[1,2,3],[4,5,6]] Output: [] Explanation: There does not exist any integer present both in nums[0] and nums[1], so we return an empty list [].

Constraints:

1 <= nums.length <= 1000 1 <= sum(nums[i].length) <= 1000 1 <= nums[i][j] <= 1000 All the values of nums[i] are unique.

==
Problem Number: 426 URL: https://leetcode.com/problems/count-prefixes-of-a-given-string Title: 2255. Count Prefixes of a Given String Problem Description: You are given a string array words and a string s, where words[i] and s comprise only of lowercase English letters. Return the number of strings in words that are a prefix of s. A prefix of a string is a substring that occurs at the beginning of the string. A substring is a contiguous sequence of characters within a string. Example 1: Input: words = ["a","b","c","ab","bc","abc"], s = "abc" Output: 3 Explanation: The strings in words which are a prefix of s = "abc" are: "a", "ab", and "abc". Thus the number of strings in words which are a prefix of s is 3. Example 2: Input: words = ["a","a"], s = "aa" Output: 2 Explanation: Both of the strings are a prefix of s. Note that the same string can occur multiple times in words, and it should be counted each time. Constraints:

1 <= words.length <= 1000 1 <= words[i].length, s.length <= 10 words[i] and s consist of lowercase English letters only.

==
Problem Number: 427 URL: https://leetcode.com/problems/remove-digit-from-number-to-maximize-result Title: 2259. Remove Digit From Number to Maximize Result Problem Description: You are given a string number representing a positive integer and a character digit. Return the resulting string after removing exactly one occurrence of digit from number such that the value of the resulting string in decimal form is maximized. The test cases are generated such that digit occurs at least once in number. Example 1: Input: number = "123", digit = "3" Output: "12" Explanation: There is only one '3' in "123". After removing '3', the result is "12".

Example 2: Input: number = "1231", digit = "1" Output: "231" Explanation: We can remove the first '1' to get "231" or remove the second '1' to get "123". Since 231 > 123, we return "231".

Example 3: Input: number = "551", digit = "5" Output: "51" Explanation: We can remove either the first or second '5' from "551". Both result in the string "51".

Constraints:

2 <= number.length <= 100 number consists of digits from '1' to '9'. digit is a digit from '1' to '9'. digit occurs at least once in number.

==
Problem Number: 428 URL: https://leetcode.com/problems/largest-3-same-digit-number-in-string Title: 2264. Largest 3-Same-Digit Number in String Problem Description: You are given a string num representing a large integer. An integer is good if it meets the following conditions:

It is a substring of num with length 3. It consists of only one unique digit.

Return the maximum good integer as a string or an empty string ”” if no such integer exists. Note:

A substring is a contiguous sequence of characters within a string. There may be leading zeroes in num or a good integer.

Example 1: Input: num = ”6777133339” Output: ”777” Explanation: There are two distinct good integers: ”777” and ”333”. ”777” is the largest, so we return ”777”.

Example 2: Input: num = ”2300019” Output: ”000” Explanation: ”000” is the only good integer.

Example 3: Input: num = ”42352338” Output: ”” Explanation: No substring of length 3 consists of only one unique digit. Therefore, there are no good integers.

Constraints:

3 <= num.length <= 1000 num only consists of digits.

==

Problem Number: 429 URL: https://leetcode.com/problems/find-the-k-beauty-of-a-number Title: 2269. Find the K-Beauty of a Number Problem Description: The k-beauty of an integer num is defined as the number of substrings of num when it is read as a string that meet the following conditions:

It has a length of k. It is a divisor of num.

Given integers num and k, return the k-beauty of num. Note:

Leading zeros are allowed. 0 is not a divisor of any value.

A substring is a contiguous sequence of characters in a string. Example 1: Input: num = 240, k = 2 Output: 2 Explanation: The following are the substrings of num of length k: - ”24” from ”240”: 24 is a divisor of 240. - ”40” from ”240”: 40 is a divisor of 240. Therefore, the k-beauty is 2.

Example 2: Input: num = 430043, k = 2 Output: 2 Explanation: The following are the substrings of num of length k: - ”43” from ”430043”: 43 is a divisor of 430043. - ”30” from ”430043”: 30 is not a divisor of 430043. - ”00” from ”430043”: 0 is not a divisor of 430043. - ”04” from ”430043”: 4 is not a divisor of 430043. - ”43” from ”430043”: 43 is a divisor of 430043. Therefore, the k-beauty is 2.

Constraints:

1 <= num <= 109 1 <= k <= num.length (taking num as a string)

==

Problem Number: 430 URL: https://leetcode.com/problems/find-resultant-array-after-removing-anagrams Title: 2273. Find Resultant Array After Removing Anagrams Problem Description: You are given a 0-indexed string array words, where words[i] consists of lowercase English letters. In one operation, select any index i such that 0 < i < words.length and words[i - 1] and words[i] are anagrams, and delete words[i] from words. Keep performing this operation as long as you can select an index that satisfies the conditions. Return words after performing all operations. It can be shown that selecting the indices for each operation in any arbitrary order will lead to the same result. An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase using all the original letters exactly once. For example, "dacb" is an anagram of "abdc". Example 1: Input: words = ["abba","baba","bbaa","cd","cd"] Output: ["abba","cd"] Explanation: One of the ways we can obtain the resultant array is by using the following operations: - Since words[2] = "bbaa" and words[1] = "baba" are anagrams, we choose index 2 and delete words[2]. Now words = ["abba","baba","cd","cd"]. - Since words[1] = "baba" and words[0] = "abba" are anagrams, we choose index 1 and delete words[1]. Now words = ["abba","cd","cd"]. - Since words[2] = "cd" and words[1] = "cd" are anagrams, we choose index 2 and delete words[2]. Now words = ["abba","cd"]. We can no longer perform any operations, so ["abba","cd"] is the final answer. Example 2: Input: words = ["a","b","c","d","e"] Output: ["a","b","c","d","e"] Explanation: No two adjacent strings in words are anagrams of each other, so no operations are performed. Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 10 words[i] consists of lowercase English letters.

==
Problem Number: 431 URL: https://leetcode.com/problems/percentage-of-letter-in-string Title: 2278. Percentage of Letter in String Problem Description: Given a string s and a character letter, return the percentage of characters in s that equal letter rounded down to the nearest whole percent. Example 1: Input: s = "foobar", letter = "o" Output: 33 Explanation: The percentage of characters in s that equal the letter 'o' is 2 / 6 * 100% = 33% when rounded down, so we return 33.

Example 2: Input: s = "jjjj", letter = "k" Output: 0 Explanation: The percentage of characters in s that equal the letter 'k' is 0%, so we return 0. Constraints:

1 <= s.length <= 100 s consists of lowercase English letters. letter is a lowercase English letter.

==
Problem Number: 432 URL: https://leetcode.com/problems/check-if-number-has-equal-digit-count-and-digit-value Title: 2283. Check if Number Has Equal Digit Count and Digit Value Problem Description: You are given a 0-indexed string num of length n consisting of digits. Return true if for every index i in the range 0 <= i < n, the digit i occurs num[i] times in num, otherwise return false. Example 1: Input: num = "1210" Output: true Explanation: num[0] = '1'. The digit 0 occurs once in num. num[1] = '2'. The digit 1 occurs twice in num. num[2] = '1'. The digit 2 occurs once in num. num[3] = '0'. The digit 3

occurs zero times in num. The condition holds true for every index in "1210", so return true.

Example 2: Input: num = "030" Output: false Explanation: num[0] = '0'. The digit 0 should occur zero times, but actually occurs twice in num. num[1] = '3'. The digit 1 should occur three times, but actually occurs zero times in num. num[2] = '0'. The digit 2 occurs zero times in num. The indices 0 and 1 both violate the condition, so return false.

Constraints:

n == num.length 1 <= n <= 10 num consists of digits.

==
Problem Number: 433 URL: https://leetcode.com/problems/rearrange-characters-to-make-target-string Title: 2287. Rearrange Characters to Make Target String Problem Description: You are given two 0-indexed strings s and target. You can take some letters from s and rearrange them to form new strings. Return the maximum number of copies of target that can be formed by taking letters from s and rearranging them. Example 1: Input: s = "ilovecodingonleetcode", target = "code" Output: 2 Explanation: For the first copy of "code", take the letters at indices 4, 5, 6, and 7. For the second copy of "code", take the letters at indices 17, 18, 19, and 20. The strings that are formed are "ecod" and "code" which can both be rearranged into "code". We can make at most two copies of "code", so we return 2.

Example 2: Input: s = "abcba", target = "abc" Output: 1 Explanation: We can make one copy of "abc" by taking the letters at indices 0, 1, and 2. We can make at most one copy of "abc", so we return 1. Note that while there is an extra 'a' and 'b' at indices 3 and 4, we cannot reuse the letter 'c' at index 2, so we cannot make a second copy of "abc".

Example 3: Input: s = "abbaccaddaeea", target = "aaaaa" Output: 1 Explanation: We can make one copy of "aaaaa" by taking the letters at indices 0, 3, 6, 9, and 12. We can make at most one copy of "aaaaa", so we return 1.

Constraints:

1 <= s.length <= 100 1 <= target.length <= 10 s and target consist of lowercase English letters.

==
Problem Number: 434 URL: https://leetcode.com/problems/min-max-game Title: 2293. Min Max Game Problem Description: You are given a 0-indexed integer array nums whose length is a power of 2. Apply the following algorithm on nums:

Let n be the length of nums. If n == 1, end the process. Otherwise, create a new 0-indexed integer array newNums of length n / 2. For every even index i where 0 <= i < n / 2, assign the value of newNums[i] as min(nums[2 * i], nums[2 * i + 1]). For every odd index i where 0 <= i < n / 2, assign the value of newNums[i] as max(nums[2 * i], nums[2 * i + 1]). Replace the array nums with newNums. Repeat the entire process starting from step 1.

Return the last number that remains in nums after applying the algorithm. Example 1:

Input: nums = [1,3,5,2,4,8,2,2] Output: 1 Explanation: The following arrays are the results of applying the algorithm repeatedly. First: nums = [1,5,4,2] Second: nums = [1,4] Third: nums = [1] 1 is the last remaining number, so we return 1.

Example 2: Input: nums = [3] Output: 3 Explanation: 3 is already the last remaining number, so we return 3.

Constraints:

1 <= nums.length <= 1024 1 <= nums[i] <= 109 nums.length is a power of 2.

==

Problem Number: 435 URL: https://leetcode.com/problems/strong-password-checker-ii Title: 2299. Strong Password Checker II Problem Description: A password is said to be strong if it satisfies all the following criteria:

It has at least 8 characters. It contains at least one lowercase letter. It contains at least one uppercase letter. It contains at least one digit. It contains at least one special character. The special characters are the characters in the following string: "!@#$%^&*()-+". It does not contain 2 of the same character in adjacent positions (i.e., "aab" violates this condition, but "aba" does not).

Given a string password, return true if it is a strong password. Otherwise, return false. Example 1: Input: password = "IloveLe3tcode!" Output: true Explanation: The password meets all the requirements. Therefore, we return true.

Example 2: Input: password = "Me+You--IsMyDream" Output: false Explanation: The password does not contain a digit and also contains 2 of the same character in adjacent positions. Therefore, we return false.

Example 3: Input: password = "1aB!" Output: false Explanation: The password does not meet the length requirement. Therefore, we return false. Constraints:

1 <= password.length <= 100 password consists of letters, digits, and special characters: "!@#$%^&*()-+".

==

Problem Number: 436 URL: https://leetcode.com/problems/calculate-amount-paid-in-taxes Title: 2303. Calculate Amount Paid in Taxes Problem Description: You are given a 0-indexed 2D integer array brackets where brackets[i] = [upperi, percenti] means that the ith tax bracket has an upper bound of upperi and is taxed at a rate of percenti. The brackets are sorted by upper bound (i.e. upperi-1 < upperi for 0 < i < brackets.length). Tax is calculated as follows:

The first upper0 dollars earned are taxed at a rate of percent0. The next upper1 - upper0 dollars earned are taxed at a rate of percent1. The next upper2 - upper1 dollars earned are taxed at a rate of percent2. And so on.

You are given an integer income representing the amount of money you earned. Return the amount of money that you have to pay in taxes. Answers within 10-5 of the actual answer will be accepted. Example 1: Input: brackets = [[3,50],[7,10],[12,25]], income = 10 Output: 2.65000 Explanation: Based on your income, you have 3 dollars in the 1st tax bracket, 4 dollars in the 2nd tax bracket, and 3 dollars in the 3rd tax bracket. The tax rate for the three tax brackets is 50%, 10%, and 25%, respectively. In total, you pay $3 * 50% + $4 * 10% + $3 * 25% = $2.65 in taxes.

Example 2: Input: brackets = [[1,0],[4,25],[5,50]], income = 2 Output: 0.25000 Explanation: Based on your income, you have 1 dollar in the 1st tax bracket and 1 dollar in the 2nd tax bracket. The tax rate for the two tax brackets is 0% and 25%, respectively. In total, you pay $1 * 0% + $1 * 25% = $0.25 in taxes.

Example 3: Input: brackets = [[2,50]], income = 0 Output: 0.00000 Explanation: You have no income to tax, so you have to pay a total of $0 in taxes.

Constraints:

1 <= brackets.length <= 100 1 <= upperi <= 1000 0 <= percenti <= 100 0 <= income <= 1000 upperi is sorted in ascending order. All the values of upperi are unique. The upper bound of the last tax bracket is greater than or equal to income.

==

Problem Number: 437 URL: https://leetcode.com/problems/greatest-english-letter-in-upper-and-lower-case Title: 2309. Greatest English Letter in Upper and Lower Case Problem Description: Given a string of English letters s, return the greatest English letter which occurs as both a lowercase and uppercase letter in s. The returned letter should be in uppercase. If no such letter exists, return an empty string. An English letter b is greater than another letter a if b appears after a in the English alphabet. Example 1: Input: s = "lEeTcOdE" Output: "E" Explanation: The letter 'E' is the only letter to appear in both lower and upper case.

Example 2: Input: s = "arRAzFif" Output: "R" Explanation: The letter 'R' is the greatest letter to appear in both lower and upper case. Note that 'A' and 'F' also appear in both lower and upper case, but 'R' is greater than 'F' or 'A'.

Example 3: Input: s = "AbCdEfGhIjK" Output: "" Explanation: There is no letter that appears in both lower and upper case.

Constraints:

1 <= s.length <= 1000 s consists of lowercase and uppercase English letters.

==

Problem Number: 438 URL: https://leetcode.com/problems/count-asterisks Title: 2315. Count Asterisks Problem Description: You are given a string s, where every two consecutive vertical bars '|' are grouped into a pair. In other words, the 1st and 2nd '|' make a pair, the 3rd and 4th '|' make a pair, and so forth. Return the number of '*' in s, excluding the '*' between each pair of '|'. Note that each '|' will belong to exactly one pair. Example 1: Input: s = "l|*e*et|c**o|*de|" Output: 2 Explanation: The considered characters are underlined: "l|*e*et|c**o|*de|". The characters between the first and second '|' are excluded from the answer. Also, the characters between the third and fourth '|' are excluded from the answer. There are 2 asterisks considered. Therefore, we return 2. Example 2: Input: s = "iamprogrammer" Output: 0 Explanation: In this example, there are no asterisks in s. Therefore, we return 0.

Example 3: Input: s = "yo|uar|e**|b|e***au|tifu|l" Output: 5 Explanation: The considered characters are underlined: "yo|uar|e**|b|e***au|tifu|l". There are 5 asterisks considered. Therefore, we return 5. Constraints:

1 <= s.length <= 1000 s consists of lowercase English letters, vertical bars '|', and asterisks '*'. s contains an even number of vertical bars '|'.

==
Problem Number: 439 URL: https://leetcode.com/problems/check-if-matrix-is-x-matrix Title: 2319. Check if Matrix Is X-Matrix Problem Description: A square matrix is said to be an X-Matrix if both of the following conditions hold:

All the elements in the diagonals of the matrix are non-zero. All other elements are 0.

Given a 2D integer array grid of size n x n representing a square matrix, return true if grid is an X-Matrix. Otherwise, return false. Example 1:

Input: grid = [[2,0,0,1],[0,3,1,0],[0,5,2,0],[4,0,0,2]] Output: true Explanation: Refer to the diagram above. An X-Matrix should have the green elements (diagonals) be non-zero and the red elements be 0. Thus, grid is an X-Matrix.

Example 2:

Input: grid = [[5,7,0],[0,3,1],[0,5,0]] Output: false Explanation: Refer to the diagram above. An X-Matrix should have the green elements (diagonals) be non-zero and the red elements be 0. Thus, grid is not an X-Matrix.

Constraints:

n == grid.length == grid[i].length 3 <= n <= 100 0 <= grid[i][j] <= 105

==
Problem Number: 440 URL: https://leetcode.com/problems/decode-the-message Title: 2325. Decode the Message Problem Description: You are given

the strings key and message, which represent a cipher key and a secret message, respectively. The steps to decode message are as follows:

Use the first appearance of all 26 lowercase English letters in key as the order of the substitution table. Align the substitution table with the regular English alphabet. Each letter in message is then substituted using the table. Spaces ' ' are transformed to themselves.

For example, given key = "happy boy" (actual key would have at least one instance of each letter in the alphabet), we have the partial substitution table of ('h' -> 'a', 'a' -> 'b', 'p' -> 'c', 'y' -> 'd', 'b' -> 'e', 'o' -> 'f').

Return the decoded message. Example 1:

Input: key = "the quick brown fox jumps over the lazy dog", message = "vkbs bs t suepuv" Output: "this is a secret" Explanation: The diagram above shows the substitution table. It is obtained by taking the first appearance of each letter in "the quick brown fox jumps over the lazy dog".

Example 2:

Input: key = "eljuxhpwnyrdgtqkviszcfmabo", message = "zwx hnfx lqantp mnoeius ycgk vcnjrdb" Output: "the five boxing wizards jump quickly" Explanation: The diagram above shows the substitution table. It is obtained by taking the first appearance of each letter in "eljuxhpwnyrdgtqkviszcfmabo".

Constraints:

26 <= key.length <= 2000 key consists of lowercase English letters and ' '. key contains every letter in the English alphabet ('a' to 'z') at least once. 1 <= message.length <= 2000 message consists of lowercase English letters and ' '.

==

Problem Number: 441 URL: https://leetcode.com/problems/evaluate-boolean-binary-tree Title: 2331. Evaluate Boolean Binary Tree Problem Description: You are given the root of a full binary tree with the following properties:

Leaf nodes have either the value 0 or 1, where 0 represents False and 1 represents True. Non-leaf nodes have either the value 2 or 3, where 2 represents the boolean OR and 3 represents the boolean AND.

The evaluation of a node is as follows:

If the node is a leaf node, the evaluation is the value of the node, i.e. True or False. Otherwise, evaluate the node's two children and apply the boolean operation of its value with the children's evaluations.

Return the boolean result of evaluating the root node. A full binary tree is a binary tree where each node has either 0 or 2 children. A leaf node is a node that has zero children. Example 1:

Input: root = [2,1,3,null,null,0,1] Output: true Explanation: The above diagram illustrates the evaluation process. The AND node evaluates to False AND True

= False. The OR node evaluates to True OR False = True. The root node evaluates to True, so we return true. Example 2: Input: root = [0] Output: false Explanation: The root node is a leaf node and it evaluates to false, so we return false.

Constraints:

The number of nodes in the tree is in the range [1, 1000]. 0 <= Node.val <= 3 Every node has either 0 or 2 children. Leaf nodes have a value of 0 or 1. Non-leaf nodes have a value of 2 or 3.

==
Problem Number: 442 URL: https://leetcode.com/problems/minimum-amount-of-time-to-fill-cups Title: 2335. Minimum Amount of Time to Fill Cups Problem Description: You have a water dispenser that can dispense cold, warm, and hot water. Every second, you can either fill up 2 cups with different types of water, or 1 cup of any type of water. You are given a 0-indexed integer array amount of length 3 where amount[0], amount[1], and amount[2] denote the number of cold, warm, and hot water cups you need to fill respectively. Return the minimum number of seconds needed to fill up all the cups. Example 1: Input: amount = [1,4,2] Output: 4 Explanation: One way to fill up the cups is: Second 1: Fill up a cold cup and a warm cup. Second 2: Fill up a warm cup and a hot cup. Second 3: Fill up a warm cup and a hot cup. Second 4: Fill up a warm cup. It can be proven that 4 is the minimum number of seconds needed.

Example 2: Input: amount = [5,4,4] Output: 7 Explanation: One way to fill up the cups is: Second 1: Fill up a cold cup, and a hot cup. Second 2: Fill up a cold cup, and a warm cup. Second 3: Fill up a cold cup, and a warm cup. Second 4: Fill up a warm cup, and a hot cup. Second 5: Fill up a cold cup, and a hot cup. Second 6: Fill up a cold cup, and a warm cup. Second 7: Fill up a hot cup.

Example 3: Input: amount = [5,0,0] Output: 5 Explanation: Every second, we fill up a cold cup.

Constraints:

amount.length == 3 0 <= amount[i] <= 100

==
Problem Number: 443 URL: https://leetcode.com/problems/maximum-number-of-pairs-in-array Title: 2341. Maximum Number of Pairs in Array Problem Description: You are given a 0-indexed integer array nums. In one operation, you may do the following:

Choose two integers in nums that are equal. Remove both integers from nums, forming a pair.

The operation is done on nums as many times as possible. Return a 0-indexed integer array answer of size 2 where answer[0] is the number of pairs that are formed and answer[1] is the number of leftover integers in nums after doing the operation as many times as possible. Example 1: Input: nums = [1,3,2,1,3,2,2] Output: [3,1] Explanation: Form a pair with nums[0] and nums[3] and remove them from nums. Now, nums = [3,2,3,2,2]. Form a pair with nums[0] and nums[2] and remove them from nums. Now, nums = [2,2,2]. Form a pair with nums[0] and nums[1] and remove them from nums. Now, nums = [2]. No more pairs can be formed. A total of 3 pairs have been formed, and there is 1 number leftover in nums.

Example 2: Input: nums = [1,1] Output: [1,0] Explanation: Form a pair with nums[0] and nums[1] and remove them from nums. Now, nums = []. No more pairs can be formed. A total of 1 pair has been formed, and there are 0 numbers leftover in nums.

Example 3: Input: nums = [0] Output: [0,1] Explanation: No pairs can be formed, and there is 1 number leftover in nums.

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 100

==

Problem Number: 444 URL: https://leetcode.com/problems/best-poker-hand Title: 2347. Best Poker Hand Problem Description: You are given an integer array ranks and a character array suits. You have 5 cards where the ith card has a rank of ranks[i] and a suit of suits[i]. The following are the types of poker hands you can make from best to worst:

"Flush": Five cards of the same suit. "Three of a Kind": Three cards of the same rank. "Pair": Two cards of the same rank. "High Card": Any single card.

Return a string representing the best type of poker hand you can make with the given cards. Note that the return values are case-sensitive. Example 1: Input: ranks = [13,2,3,1,9], suits = ["a","a","a","a","a"] Output: "Flush" Explanation: The hand with all the cards consists of 5 cards with the same suit, so we have a "Flush".

Example 2: Input: ranks = [4,4,2,4,4], suits = ["d","a","a","b","c"] Output: "Three of a Kind" Explanation: The hand with the first, second, and fourth card consists of 3 cards with the same rank, so we have a "Three of a Kind". Note that we could also make a "Pair" hand but "Three of a Kind" is a better hand. Also note that other cards could be used to make the "Three of a Kind" hand. Example 3: Input: ranks = [10,10,2,12,9], suits = ["a","b","c","a","d"] Output: "Pair" Explanation: The hand with the first and second card consists of 2 cards with the same rank, so we have a "Pair". Note that we cannot make a "Flush" or a "Three of a Kind".

Constraints:

ranks.length == suits.length == 5 1 <= ranks[i] <= 13 'a' <= suits[i] <= 'd' No two cards have the same rank and suit.

==
Problem Number: 445 URL: https://leetcode.com/problems/first-letter-to-appear-twice Title: 2351. First Letter to Appear Twice Problem Description: Given a string s consisting of lowercase English letters, return the first letter to appear twice. Note:

A letter a appears twice before another letter b if the second occurrence of a is before the second occurrence of b. s will contain at least one letter that appears twice.

Example 1: Input: s = "abccbaacz" Output: "c" Explanation: The letter 'a' appears on the indexes 0, 5 and 6. The letter 'b' appears on the indexes 1 and 4. The letter 'c' appears on the indexes 2, 3 and 7. The letter 'z' appears on the index 8. The letter 'c' is the first letter to appear twice, because out of all the letters the index of its second occurrence is the smallest.

Example 2: Input: s = "abcdd" Output: "d" Explanation: The only letter that appears twice is 'd' so we return 'd'.

Constraints:

2 <= s.length <= 100 s consists of lowercase English letters. s has at least one repeated letter.

==
Problem Number: 446 URL: https://leetcode.com/problems/make-array-zero-by-subtracting-equal-amounts Title: 2357. Make Array Zero by Subtracting Equal Amounts Problem Description: You are given a non-negative integer array nums. In one operation, you must:

Choose a positive integer x such that x is less than or equal to the smallest non-zero element in nums. Subtract x from every positive element in nums.

Return the minimum number of operations to make every element in nums equal to 0. Example 1: Input: nums = [1,5,0,3,5] Output: 3 Explanation: In the first operation, choose x = 1. Now, nums = [0,4,0,2,4]. In the second operation, choose x = 2. Now, nums = [0,2,0,0,2]. In the third operation, choose x = 2. Now, nums = [0,0,0,0,0].

Example 2: Input: nums = [0] Output: 0 Explanation: Each element in nums is already 0 so no operations are needed.

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 100

==
Problem Number: 447 URL: https://leetcode.com/problems/merge-similar-items Title: 2363. Merge Similar Items Problem Description: You are given two 2D integer arrays, items1 and items2, representing two sets of items. Each array items has the following properties:

items[i] = [valuei, weighti] where valuei represents the value and weighti represents the weight of the ith item. The value of each item in items is unique.

Return a 2D integer array ret where ret[i] = [valuei, weighti], with weighti being the sum of weights of all items with value valuei. Note: ret should be returned in ascending order by value. Example 1: Input: items1 = [[1,1],[4,5],[3,8]], items2 = [[3,1],[1,5]] Output: [[1,6],[3,9],[4,5]] Explanation: The item with value = 1 occurs in items1 with weight = 1 and in items2 with weight = 5, total weight = 1 + 5 = 6. The item with value = 3 occurs in items1 with weight = 8 and in items2 with weight = 1, total weight = 8 + 1 = 9. The item with value = 4 occurs in items1 with weight = 5, total weight = 5. Therefore, we return [[1,6],[3,9],[4,5]].

Example 2: Input: items1 = [[1,1],[3,2],[2,3]], items2 = [[2,1],[3,2],[1,3]] Output: [[1,4],[2,4],[3,4]] Explanation: The item with value = 1 occurs in items1 with weight = 1 and in items2 with weight = 3, total weight = 1 + 3 = 4. The item with value = 2 occurs in items1 with weight = 3 and in items2 with weight = 1, total weight = 3 + 1 = 4. The item with value = 3 occurs in items1 with weight = 2 and in items2 with weight = 2, total weight = 2 + 2 = 4. Therefore, we return [[1,4],[2,4],[3,4]]. Example 3: Input: items1 = [[1,3],[2,2]], items2 = [[7,1],[2,2],[1,4]] Output: [[1,7],[2,4],[7,1]] Explanation: The item with value = 1 occurs in items1 with weight = 3 and in items2 with weight = 4, total weight = 3 + 4 = 7. The item with value = 2 occurs in items1 with weight = 2 and in items2 with weight = 2, total weight = 2 + 2 = 4. The item with value = 7 occurs in items2 with weight = 1, total weight = 1. Therefore, we return [[1,7],[2,4],[7,1]].

Constraints:

1 <= items1.length, items2.length <= 1000 items1[i].length == items2[i].length == 2 1 <= valuei, weighti <= 1000 Each valuei in items1 is unique. Each valuei in items2 is unique.

==
Problem Number: 448 URL: https://leetcode.com/problems/number-of-arithmetic-triplets Title: 2367. Number of Arithmetic Triplets Problem Description: You are given a 0-indexed, strictly increasing integer array nums and a positive integer diff. A triplet (i, j, k) is an arithmetic triplet if the following conditions are met:

i < j < k, nums[j] - nums[i] == diff, and nums[k] - nums[j] == diff.

Return the number of unique arithmetic triplets. Example 1: Input: nums = [0,1,4,6,7,10], diff = 3 Output: 2 Explanation: (1, 2, 4) is an arithmetic triplet because both 7 - 4 == 3 and 4 - 1 == 3. (2, 4, 5) is an arithmetic triplet because both 10 - 7 == 3 and 7 - 4 == 3.

Example 2: Input: nums = [4,5,6,7,8,9], diff = 2 Output: 2 Explanation: (0, 2, 4) is an arithmetic triplet because both 8 - 6 == 2 and 6 - 4 == 2. (1, 3, 5) is an arithmetic triplet because both 9 - 7 == 2 and 7 - 5 == 2.

Constraints:

3 <= nums.length <= 200 0 <= nums[i] <= 200 1 <= diff <= 50 nums is strictly increasing.

==

Problem Number: 449 URL: https://leetcode.com/problems/largest-local-values-in-a-matrix Title: 2373. Largest Local Values in a Matrix Problem Description: You are given an n x n integer matrix grid. Generate an integer matrix maxLocal of size (n - 2) x (n - 2) such that:

maxLocal[i][j] is equal to the largest value of the 3 x 3 matrix in grid centered around row i + 1 and column j + 1.

In other words, we want to find the largest value in every contiguous 3 x 3 matrix in grid. Return the generated matrix.   Example 1:

Input: grid = [[9,9,8,1],[5,6,2,6],[8,2,6,4],[6,2,2,2]] Output: [[9,9],[8,6]] Explanation: The diagram above shows the original matrix and the generated matrix. Notice that each value in the generated matrix corresponds to the largest value of a contiguous 3 x 3 matrix in grid. Example 2:

Input: grid = [[1,1,1,1,1],[1,1,1,1,1],[1,1,2,1,1],[1,1,1,1,1],[1,1,1,1,1]] Output: [[2,2,2],[2,2,2],[2,2,2]] Explanation: Notice that the 2 is contained within every contiguous 3 x 3 matrix in grid.

Constraints:

n == grid.length == grid[i].length 3 <= n <= 100 1 <= grid[i][j] <= 100

==

Problem Number: 450 URL: https://leetcode.com/problems/minimum-recolors-to-get-k-consecutive-black-blocks Title: 2379. Minimum Recolors to Get K Consecutive Black Blocks Problem Description: You are given a 0-indexed string blocks of length n, where blocks[i] is either 'W' or 'B', representing the color of the ith block. The characters 'W' and 'B' denote the colors white and black, respectively. You are also given an integer k, which is the desired number of consecutive black blocks. In one operation, you can recolor a white block such that it becomes a black block. Return the minimum number of operations needed such that there is at least one occurrence of k consecutive black blocks.   Example 1: Input: blocks = "WBBWWBBWBW", k = 7 Output: 3 Explanation: One way to achieve 7 consecutive black blocks is to recolor the 0th, 3rd, and 4th blocks so that blocks = "BBBBBBBWBW". It can be shown that there is no way to achieve 7 consecutive black blocks in less than 3 operations. Therefore, we return 3.

Example 2: Input: blocks = "WBWBBBW", k = 2 Output: 0 Explanation: No changes need to be made, since 2 consecutive black blocks already exist. Therefore, we return 0.

Constraints:

n == blocks.length 1 <= n <= 100 blocks[i] is either 'W' or 'B'. 1 <= k <= n

==

Problem Number: 451 URL: https://leetcode.com/problems/minimum-hours-of-training-to-win-a-competition Title: 2383. Minimum Hours of Training to Win a Competition Problem Description: You are entering a competition, and are given two positive integers initialEnergy and initialExperience denoting your initial energy and initial experience respectively. You are also given two 0-indexed integer arrays energy and experience, both of length n. You will face n opponents in order. The energy and experience of the ith opponent is denoted by energy[i] and experience[i] respectively. When you face an opponent, you need to have both strictly greater experience and energy to defeat them and move to the next opponent if available. Defeating the ith opponent increases your experience by experience[i], but decreases your energy by energy[i]. Before starting the competition, you can train for some number of hours. After each hour of training, you can either choose to increase your initial experience by one, or increase your initial energy by one. Return the minimum number of training hours required to defeat all n opponents.   Example 1: Input: initialEnergy = 5, initialExperience = 3, energy = [1,4,3,2], experience = [2,6,3,1] Output: 8 Explanation: You can increase your energy to 11 after 6 hours of training, and your experience to 5 after 2 hours of training. You face the opponents in the following order: - You have more energy and experience than the 0th opponent so you win. Your energy becomes 11 - 1 = 10, and your experience becomes 5 + 2 = 7. - You have more energy and experience than the 1st opponent so you win. Your energy becomes 10 - 4 = 6, and your experience becomes 7 + 6 = 13. - You have more energy and experience than the 2nd opponent so you win. Your energy becomes 6 - 3 = 3, and your experience becomes 13 + 3 = 16. - You have more energy and experience than the 3rd opponent so you win. Your energy becomes 3 - 2 = 1, and your experience becomes 16 + 1 = 17. You did a total of 6 + 2 = 8 hours of training before the competition, so we return 8. It can be proven that no smaller answer exists.

Example 2: Input: initialEnergy = 2, initialExperience = 4, energy = [1], experience = [3] Output: 0 Explanation: You do not need any additional energy or experience to win the competition, so we return 0.

Constraints:

n == energy.length == experience.length 1 <= n <= 100 1 <= initialEnergy, initialExperience, energy[i], experience[i] <= 100

==

Problem Number: 452 URL: https://leetcode.com/problems/longest-subsequence-with-limited-sum Title: 2389. Longest Subsequence With Limited Sum Problem Description: You are given an integer array nums of length n, and an integer array queries of length m. Return an array answer of length m where answer[i] is the maximum size of a subsequence that you can take from nums such that the sum of its elements is less than or equal to queries[i]. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.   Example 1: Input: nums = [4,5,2,1], queries = [3,10,21] Output: [2,3,4] Explanation: We answer the queries as follows: - The subsequence [2,1] has a sum less than or equal to 3. It can be proven that 2 is the maximum size of such a subsequence, so answer[0] = 2. - The subsequence [4,5,1] has a sum less than or equal to 10. It can be proven that 3 is the maximum size of such a subsequence, so answer[1] = 3. - The subsequence [4,5,2,1] has a sum less than or equal to 21. It can be proven that 4 is the maximum size of such a subsequence, so answer[2] = 4.

Example 2: Input: nums = [2,3,4,5], queries = [1] Output: [0] Explanation: The empty subsequence is the only subsequence that has a sum less than or equal to 1, so answer[0] = 0.   Constraints:

n == nums.length m == queries.length 1 <= n, m <= 1000 1 <= nums[i], queries[i] <= 106

==
Problem Number: 453 URL: https://leetcode.com/problems/find-subarrays-with-equal-sum Title: 2395. Find Subarrays With Equal Sum Problem Description: Given a 0-indexed integer array nums, determine whether there exist two subarrays of length 2 with equal sum. Note that the two subarrays must begin at different indices. Return true if these subarrays exist, and false otherwise. A subarray is a contiguous non-empty sequence of elements within an array.   Example 1: Input: nums = [4,2,4] Output: true Explanation: The subarrays with elements [4,2] and [2,4] have the same sum of 6.

Example 2: Input: nums = [1,2,3,4,5] Output: false Explanation: No two subarrays of size 2 have the same sum.

Example 3: Input: nums = [0,0,0] Output: true Explanation: The subarrays [nums[0],nums[1]] and [nums[1],nums[2]] have the same sum of 0. Note that even though the subarrays have the same content, the two subarrays are considered different because they are in different positions in the original array.

Constraints:

2 <= nums.length <= 1000 -109 <= nums[i] <= 109

==
Problem Number: 454 URL: https://leetcode.com/problems/check-distances-between-same-letters Title: 2399. Check Distances Between Same Letters Problem Description: You are given a 0-indexed string s consisting of only lowercase English letters, where each letter in s appears exactly twice. You are also given a 0-indexed integer array distance of length 26. Each letter in the alphabet is numbered from 0 to 25 (i.e. 'a' -> 0, 'b' -> 1, 'c' -> 2, ... , 'z' -> 25). In a well-spaced string, the number of letters between the two occurrences of the ith letter is distance[i]. If the ith letter does not appear in s, then distance[i] can be ignored. Return true if s is a well-spaced

string, otherwise return false. Example 1: Input: s = "abaccb", distance = [1,3,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] Output: true Explanation: - 'a' appears at indices 0 and 2 so it satisfies distance[0] = 1. - 'b' appears at indices 1 and 5 so it satisfies distance[1] = 3. - 'c' appears at indices 3 and 4 so it satisfies distance[2] = 0. Note that distance[3] = 5, but since 'd' does not appear in s, it can be ignored. Return true because s is a well-spaced string.

Example 2: Input: s = "aa", distance = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] Output: false Explanation: - 'a' appears at indices 0 and 1 so there are zero letters between them. Because distance[0] = 1, s is not a well-spaced string.

Constraints:

2 <= s.length <= 52 s consists only of lowercase English letters. Each letter appears in s exactly twice. distance.length == 26 0 <= distance[i] <= 50

==

Problem Number: 455 URL: https://leetcode.com/problems/most-frequent-even-element Title: 2404. Most Frequent Even Element Problem Description: Given an integer array nums, return the most frequent even element. If there is a tie, return the smallest one. If there is no such element, return -1. Example 1: Input: nums = [0,1,2,2,4,4,1] Output: 2 Explanation: The even elements are 0, 2, and 4. Of these, 2 and 4 appear the most. We return the smallest one, which is 2. Example 2: Input: nums = [4,4,4,9,2,4] Output: 4 Explanation: 4 is the even element appears the most.

Example 3: Input: nums = [29,47,21,41,13,37,25,7] Output: -1 Explanation: There is no even element.

Constraints:

1 <= nums.length <= 2000 0 <= nums[i] <= 105

==

Problem Number: 456 URL: https://leetcode.com/problems/count-days-spent-together Title: 2409. Count Days Spent Together Problem Description: Alice and Bob are traveling to Rome for separate business meetings. You are given 4 strings arriveAlice, leaveAlice, arriveBob, and leaveBob. Alice will be in the city from the dates arriveAlice to leaveAlice (inclusive), while Bob will be in the city from the dates arriveBob to leaveBob (inclusive). Each will be a 5-character string in the format "MM-DD", corresponding to the month and day of the date. Return the total number of days that Alice and Bob are in Rome together. You can assume that all dates occur in the same calendar year, which is not a leap year. Note that the number of days per month can be represented as: [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]. Example 1: Input: arriveAlice = "08-15", leaveAlice = "08-18", arriveBob = "08-16", leaveBob = "08-19" Output: 3 Explanation: Alice will be in Rome from August 15 to August 18. Bob will be in Rome from August 16 to August 19. They are both in Rome together on August 16th, 17th, and 18th, so the answer is 3.

Example 2: Input: arriveAlice = ”10-01”, leaveAlice = ”10-31”, arriveBob = ”11-01”, leaveBob = ”12-31” Output: 0 Explanation: There is no day when Alice and Bob are in Rome together, so we return 0.

Constraints:

All dates are provided in the format ”MM-DD”. Alice and Bob’s arrival dates are earlier than or equal to their leaving dates. The given dates are valid dates of a non-leap year.

==

Problem Number: 457 URL: https://leetcode.com/problems/smallest-even-multiple Title: 2413. Smallest Even Multiple Problem Description: Given a positive integer n, return the smallest positive integer that is a multiple of both 2 and n. Example 1: Input: n = 5 Output: 10 Explanation: The smallest multiple of both 5 and 2 is 10.

Example 2: Input: n = 6 Output: 6 Explanation: The smallest multiple of both 6 and 2 is 6. Note that a number is a multiple of itself.

Constraints:

1 <= n <= 150

==

Problem Number: 458 URL: https://leetcode.com/problems/sort-the-people Title: 2418. Sort the People Problem Description: You are given an array of strings names, and an array heights that consists of distinct positive integers. Both arrays are of length n. For each index i, names[i] and heights[i] denote the name and height of the ith person. Return names sorted in descending order by the people’s heights. Example 1: Input: names = [”Mary”,”John”,”Emma”], heights = [180,165,170] Output: [”Mary”,”Emma”,”John”] Explanation: Mary is the tallest, followed by Emma and John.

Example 2: Input: names = [”Alice”,”Bob”,”Bob”], heights = [155,185,150] Output: [”Bob”,”Alice”,”Bob”] Explanation: The first Bob is the tallest, followed by Alice and the second Bob.

Constraints:

n == names.length == heights.length 1 <= n <= 103 1 <= names[i].length <= 20 1 <= heights[i] <= 105 names[i] consists of lower and upper case English letters. All the values of heights are distinct.

==

Problem Number: 459 URL: https://leetcode.com/problems/remove-letter-to-equalize-frequency Title: 2423. Remove Letter To Equalize Frequency Problem Description: You are given a 0-indexed string word, consisting of lowercase English letters. You need to select one index and remove the letter at that index from word so that the frequency of every letter present in word is equal.

Return true if it is possible to remove one letter so that the frequency of all letters in word are equal, and false otherwise. Note:

The frequency of a letter x is the number of times it occurs in the string. You must remove exactly one letter and cannot chose to do nothing.

Example 1: Input: word = "abcc" Output: true Explanation: Select index 3 and delete it: word becomes "abc" and each character has a frequency of 1.

Example 2: Input: word = "aazz" Output: false Explanation: We must delete a character, so either the frequency of "a" is 1 and the frequency of "z" is 2, or vice versa. It is impossible to make all present letters have equal frequency.

Constraints:

2 <= word.length <= 100 word consists of lowercase English letters only.

==
Problem Number: 460 URL: https://leetcode.com/problems/number-of-common-factors Title: 2427. Number of Common Factors Problem Description: Given two positive integers a and b, return the number of common factors of a and b. An integer x is a common factor of a and b if x divides both a and b. Example 1: Input: a = 12, b = 6 Output: 4 Explanation: The common factors of 12 and 6 are 1, 2, 3, 6.

Example 2: Input: a = 25, b = 30 Output: 2 Explanation: The common factors of 25 and 30 are 1, 5.

Constraints:

1 <= a, b <= 1000

==
Problem Number: 461 URL: https://leetcode.com/problems/the-employee-that-worked-on-the-longest-task Title: 2432. The Employee That Worked on the Longest Task Problem Description: There are n employees, each with a unique id from 0 to n - 1. You are given a 2D integer array logs where logs[i] = [idi, leaveTimei] where:

idi is the id of the employee that worked on the ith task, and leaveTimei is the time at which the employee finished the ith task. All the values leaveTimei are unique.

Note that the ith task starts the moment right after the (i - 1)th task ends, and the 0th task starts at time 0. Return the id of the employee that worked the task with the longest time. If there is a tie between two or more employees, return the smallest id among them. Example 1: Input: n = 10, logs = [[0,3],[2,5],[0,9],[1,15]] Output: 1 Explanation: Task 0 started at 0 and ended at 3 with 3 units of times. Task 1 started at 3 and ended at 5 with 2 units of times. Task 2 started at 5 and ended at 9 with 4 units of times. Task 3 started at 9 and ended at 15 with 6 units of times. The task with the longest time is task 3 and the employee with id 1 is the one that worked on it, so we return 1.

Example 2: Input: n = 26, logs = [[1,1],[3,7],[2,12],[7,17]] Output: 3 Explanation: Task 0 started at 0 and ended at 1 with 1 unit of times. Task 1 started at 1 and ended at 7 with 6 units of times. Task 2 started at 7 and ended at 12 with 5 units of times. Task 3 started at 12 and ended at 17 with 5 units of times. The tasks with the longest time is task 1. The employee that worked on it is 3, so we return 3.

Example 3: Input: n = 2, logs = [[0,10],[1,20]] Output: 0 Explanation: Task 0 started at 0 and ended at 10 with 10 units of times. Task 1 started at 10 and ended at 20 with 10 units of times. The tasks with the longest time are tasks 0 and 1. The employees that worked on them are 0 and 1, so we return the smallest id 0.

Constraints:

2 <= n <= 500 1 <= logs.length <= 500 logs[i].length == 2 0 <= idi <= n - 1 1 <= leaveTimei <= 500 idi != idi+1 leaveTimei are sorted in a strictly increasing order.

==
Problem Number: 462 URL: https://leetcode.com/problems/number-of-valid-clock-times Title: 2437. Number of Valid Clock Times Problem Description: You are given a string of length 5 called time, representing the current time on a digital clock in the format "hh:mm". The earliest possible time is "00:00" and the latest possible time is "23:59". In the string time, the digits represented by the ? symbol are unknown, and must be replaced with a digit from 0 to 9. Return an integer answer, the number of valid clock times that can be created by replacing every ? with a digit from 0 to 9. Example 1: Input: time = "?5:00" Output: 2 Explanation: We can replace the ? with either a 0 or 1, producing "05:00" or "15:00". Note that we cannot replace it with a 2, since the time "25:00" is invalid. In total, we have two choices.

Example 2: Input: time = "0?:0?" Output: 100 Explanation: Each ? can be replaced by any digit from 0 to 9, so we have 100 total choices.

Example 3: Input: time = "??:??" Output: 1440 Explanation: There are 24 possible choices for the hours, and 60 possible choices for the minutes. In total, we have 24 * 60 = 1440 choices.

Constraints:

time is a valid string of length 5 in the format "hh:mm". "00" <= hh <= "23" "00" <= mm <= "59" Some of the digits might be replaced with '?' and need to be replaced with digits from 0 to 9.

==
Problem Number: 463 URL: https://leetcode.com/problems/largest-positive-integer-that-exists-with-its-negative Title: 2441. Largest Positive Integer That Exists With Its Negative Problem Description: Given an integer array nums that does not contain any zeros, find the largest positive integer k such that -k

also exists in the array. Return the positive integer k. If there is no such integer, return -1.   Example 1: Input: nums = [-1,2,-3,3] Output: 3 Explanation: 3 is the only valid k we can find in the array.

Example 2: Input: nums = [-1,10,6,7,-7,1] Output: 7 Explanation: Both 1 and 7 have their corresponding negative values in the array. 7 has a larger value.

Example 3: Input: nums = [-10,8,6,7,-2,-3] Output: -1 Explanation: There is no a single valid k, we return -1.

Constraints:

1 <= nums.length <= 1000 -1000 <= nums[i] <= 1000 nums[i] != 0

==
Problem Number: 464 URL: https://leetcode.com/problems/determine-if-two-events-have-conflict Title: 2446. Determine if Two Events Have Conflict Problem Description: You are given two arrays of strings that represent two inclusive events that happened on the same day, event1 and event2, where:

event1 = [startTime1, endTime1] and event2 = [startTime2, endTime2].

Event times are valid 24 hours format in the form of HH:MM. A conflict happens when two events have some non-empty intersection (i.e., some moment is common to both events). Return true if there is a conflict between two events. Otherwise, return false.   Example 1: Input: event1 = ["01:15","02:00"], event2 = ["02:00","03:00"] Output: true Explanation: The two events intersect at time 2:00.

Example 2: Input: event1 = ["01:00","02:00"], event2 = ["01:20","03:00"] Output: true Explanation: The two events intersect starting from 01:20 to 02:00.

Example 3: Input: event1 = ["10:00","11:00"], event2 = ["14:00","15:00"] Output: false Explanation: The two events do not intersect.

Constraints:

evnet1.length == event2.length == 2. event1[i].length == event2[i].length == 5 startTime1 <= endTime1 startTime2 <= endTime2 All the event times follow the HH:MM format.

==
Problem Number: 465 URL: https://leetcode.com/problems/odd-string-difference Title: 2451. Odd String Difference Problem Description: You are given an array of equal-length strings words. Assume that the length of each string is n. Each string words[i] can be converted into a difference integer array difference[i] of length n - 1 where difference[i][j] = words[i][j+1] - words[i][j] where 0 <= j <= n - 2. Note that the difference between two letters is the difference between their positions in the alphabet i.e. the position of 'a' is 0, 'b' is 1, and 'z' is 25.

For example, for the string ”acb”, the difference integer array is [2 - 0, 1 - 2] = [2, -1].

All the strings in words have the same difference integer array, except one. You should find that string. Return the string in words that has different difference integer array. Example 1: Input: words = [”adc”,”wzy”,”abc”] Output: ”abc” Explanation: - The difference integer array of ”adc” is [3 - 0, 2 - 3] = [3, -1]. - The difference integer array of ”wzy” is [25 - 22, 24 - 25]= [3, -1]. - The difference integer array of ”abc” is [1 - 0, 2 - 1] = [1, 1]. The odd array out is [1, 1], so we return the corresponding string, ”abc”.

Example 2: Input: words = [”aaa”,”bob”,”ccc”,”ddd”] Output: ”bob” Explanation: All the integer arrays are [0, 0] except for ”bob”, which corresponds to [13, -13].

Constraints:

3 <= words.length <= 100 n == words[i].length 2 <= n <= 20 words[i] consists of lowercase English letters.

==
Problem Number: 466 URL: https://leetcode.com/problems/average-value-of-even-numbers-that-are-divisible-by-three Title: 2455. Average Value of Even Numbers That Are Divisible by Three Problem Description: Given an integer array nums of positive integers, return the average value of all even integers that are divisible by 3. Note that the average of n elements is the sum of the n elements divided by n and rounded down to the nearest integer. Example 1: Input: nums = [1,3,6,10,12,15] Output: 9 Explanation: 6 and 12 are even numbers that are divisible by 3. (6 + 12) / 2 = 9.

Example 2: Input: nums = [1,2,4,7,10] Output: 0 Explanation: There is no single number that satisfies the requirement, so return 0.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 1000

==
Problem Number: 467 URL: https://leetcode.com/problems/apply-operations-to-an-array Title: 2460. Apply Operations to an Array Problem Description: You are given a 0-indexed array nums of size n consisting of non-negative integers. You need to apply n - 1 operations to this array where, in the ith operation (0-indexed), you will apply the following on the ith element of nums:

If nums[i] == nums[i + 1], then multiply nums[i] by 2 and set nums[i + 1] to 0. Otherwise, you skip this operation.

After performing all the operations, shift all the 0’s to the end of the array.

For example, the array [1,0,2,0,0,1] after shifting all its 0’s to the end, is [1,2,1,0,0,0].

Return the resulting array. Note that the operations are applied sequentially, not all at once. Example 1: Input: nums = [1,2,2,1,1,0] Output: [1,4,2,0,0,0] Explanation: We do the following operations: - i = 0: nums[0] and nums[1] are not equal, so we skip this operation. - i = 1: nums[1] and nums[2] are equal, we multiply nums[1] by 2 and change nums[2] to 0. The array becomes [1,4,0,1,1,0]. - i = 2: nums[2] and nums[3] are not equal, so we skip this operation. - i = 3: nums[3] and nums[4] are equal, we multiply nums[3] by 2 and change nums[4] to 0. The array becomes [1,4,0,2,0,0]. - i = 4: nums[4] and nums[5] are equal, we multiply nums[4] by 2 and change nums[5] to 0. The array becomes [1,4,0,2,0,0]. After that, we shift the 0's to the end, which gives the array [1,4,2,0,0,0].

Example 2: Input: nums = [0,1] Output: [1,0] Explanation: No operation can be applied, we just shift the 0 to the end.

Constraints:

2 <= nums.length <= 2000 0 <= nums[i] <= 1000

==

Problem Number: 468 URL: https://leetcode.com/problems/number-of-distinct-averages Title: 2465. Number of Distinct Averages Problem Description: You are given a 0-indexed integer array nums of even length. As long as nums is not empty, you must repetitively:

Find the minimum number in nums and remove it. Find the maximum number in nums and remove it. Calculate the average of the two removed numbers.

The average of two numbers a and b is (a + b) / 2.

For example, the average of 2 and 3 is (2 + 3) / 2 = 2.5.

Return the number of distinct averages calculated using the above process. Note that when there is a tie for a minimum or maximum number, any can be removed. Example 1: Input: nums = [4,1,4,0,3,5] Output: 2 Explanation: 1. Remove 0 and 5, and the average is (0 + 5) / 2 = 2.5. Now, nums = [4,1,4,3]. 2. Remove 1 and 4. The average is (1 + 4) / 2 = 2.5, and nums = [4,3]. 3. Remove 3 and 4, and the average is (3 + 4) / 2 = 3.5. Since there are 2 distinct numbers among 2.5, 2.5, and 3.5, we return 2.

Example 2: Input: nums = [1,100] Output: 1 Explanation: There is only one average to be calculated after removing 1 and 100, so we return 1.

Constraints:

2 <= nums.length <= 100 nums.length is even. 0 <= nums[i] <= 100

==

Problem Number: 469 URL: https://leetcode.com/problems/convert-the-temperature Title: 2469. Convert the Temperature Problem Description: You are given a non-negative floating point number rounded to two decimal places celsius, that denotes the temperature in Celsius. You should convert Celsius into Kelvin and Fahrenheit and return it as an array ans = [kelvin, fahrenheit]. Return the array ans. Answers within 10-5 of the actual answer will be accepted. Note that:

Kelvin = Celsius + 273.15 Fahrenheit = Celsius * 1.80 + 32.00

Example 1: Input: celsius = 36.50 Output: [309.65000,97.70000] Explanation: Temperature at 36.50 Celsius converted in Kelvin is 309.65 and converted in Fahrenheit is 97.70.

Example 2: Input: celsius = 122.11 Output: [395.26000,251.79800] Explanation: Temperature at 122.11 Celsius converted in Kelvin is 395.26 and converted in Fahrenheit is 251.798.

Constraints:

0 <= celsius <= 1000

==

Problem Number: 470 URL: https://leetcode.com/problems/number-of-unequal-triplets-in-array Title: 2475. Number of Unequal Triplets in Array Problem Description: You are given a 0-indexed array of positive integers nums. Find the number of triplets (i, j, k) that meet the following conditions:

0 <= i < j < k < nums.length nums[i], nums[j], and nums[k] are pairwise distinct.

In other words, nums[i] != nums[j], nums[i] != nums[k], and nums[j] != nums[k].

Return the number of triplets that meet the conditions. Example 1: Input: nums = [4,4,2,4,3] Output: 3 Explanation: The following triplets meet the conditions: - (0, 2, 4) because 4 != 2 != 3 - (1, 2, 4) because 4 != 2 != 3 - (2, 3, 4) because 2 != 4 != 3 Since there are 3 triplets, we return 3. Note that (2, 0, 4) is not a valid triplet because 2 > 0.

Example 2: Input: nums = [1,1,1,1,1] Output: 0 Explanation: No triplets meet the conditions so we return 0.

Constraints:

3 <= nums.length <= 100 1 <= nums[i] <= 1000

==

Problem Number: 471 URL: https://leetcode.com/problems/minimum-cuts-to-divide-a-circle Title: 2481. Minimum Cuts to Divide a Circle Problem Description: A valid cut in a circle can be:

A cut that is represented by a straight line that touches two points on the edge of the circle and passes through its center, or A cut that is represented by a straight line that touches one point on the edge of the circle and its center.

Some valid and invalid cuts are shown in the figures below.

Given the integer n, return the minimum number of cuts needed to divide a circle into n equal slices. Example 1:

Input: n = 4 Output: 2 Explanation: The above figure shows how cutting the circle twice through the middle divides it into 4 equal slices.

Example 2:

Input: n = 3 Output: 3 Explanation: At least 3 cuts are needed to divide the circle into 3 equal slices. It can be shown that less than 3 cuts cannot result in 3 slices of equal size and shape. Also note that the first cut will not divide the circle into distinct parts.

Constraints:

1 <= n <= 100

==
Problem Number: 472 URL: https://leetcode.com/problems/find-the-pivot-integer Title: 2485. Find the Pivot Integer Problem Description: Given a positive integer n, find the pivot integer x such that:

The sum of all elements between 1 and x inclusively equals the sum of all elements between x and n inclusively.

Return the pivot integer x. If no such integer exists, return -1. It is guaranteed that there will be at most one pivot index for the given input. Example 1: Input: n = 8 Output: 6 Explanation: 6 is the pivot integer since: 1 + 2 + 3 + 4 + 5 + 6 = 6 + 7 + 8 = 21.

Example 2: Input: n = 1 Output: 1 Explanation: 1 is the pivot integer since: 1 = 1.

Example 3: Input: n = 4 Output: -1 Explanation: It can be proved that no such integer exist.

Constraints:

1 <= n <= 1000

==
Problem Number: 473 URL: https://leetcode.com/problems/circular-sentence Title: 2490. Circular Sentence Problem Description: A sentence is a list of words that are separated by a single space with no leading or trailing spaces.

For example, "Hello World", "HELLO", "hello world hello world" are all sentences.

Words consist of only uppercase and lowercase English letters. Uppercase and lowercase English letters are considered different. A sentence is circular if:

The last character of a word is equal to the first character of the next word. The last character of the last word is equal to the first character of the first word.

For example, "leetcode exercises sound delightful", "eetcode", "leetcode eats soul" are all circular sentences. However, "Leetcode is cool", "happy Leetcode", "Leetcode" and "I like Leetcode" are not circular sentences. Given a string sentence, return true if it is circular. Otherwise, return false. Example 1: Input: sentence = "leetcode exercises sound delightful" Output: true Explanation: The words in sentence are ["leetcode", "exercises", "sound", "delightful"]. - leetcode's last character is equal to exercises's first character. - exercises's last character is equal to sound's first character. - sound's last character is equal to delightful's first character. - delightful's last character is equal to leetcode's first character. The sentence is circular. Example 2: Input: sentence = "eetcode" Output: true Explanation: The words in sentence are ["eetcode"]. - eetcode's last character is equal to eetcode's first character. The sentence is circular. Example 3: Input: sentence = "Leetcode is cool" Output: false Explanation: The words in sentence are ["Leetcode", "is", "cool"]. - Leetcode's last character is not equal to is's first character. The sentence is not circular. Constraints:

1 <= sentence.length <= 500 sentence consist of only lowercase and uppercase English letters and spaces. The words in sentence are separated by a single space. There are no leading or trailing spaces.

==
Problem Number: 474 URL: https://leetcode.com/problems/maximum-value-of-a-string-in-an-array Title: 2496. Maximum Value of a String in an Array Problem Description: The value of an alphanumeric string can be defined as:

The numeric representation of the string in base 10, if it comprises of digits only. The length of the string, otherwise.

Given an array strs of alphanumeric strings, return the maximum value of any string in strs. Example 1: Input: strs = ["alic3","bob","3","4","00000"] Output: 5 Explanation: - "alic3" consists of both letters and digits, so its value is its length, i.e. 5. - "bob" consists only of letters, so its value is also its length, i.e. 3. - "3" consists only of digits, so its value is its numeric equivalent, i.e. 3. - "4" also consists only of digits, so its value is 4. - "00000" consists only of digits, so its value is 0. Hence, the maximum value is 5, of "alic3".

Example 2: Input: strs = ["1","01","001","0001"] Output: 1 Explanation: Each string in the array has value 1. Hence, we return 1.

Constraints:

1 <= strs.length <= 100 1 <= strs[i].length <= 9 strs[i] consists of only lowercase English letters and digits.

==
Problem Number: 475 URL: https://leetcode.com/problems/delete-greatest-value-in-each-row Title: 2500. Delete Greatest Value in Each Row Problem Description: You are given an m x n matrix grid consisting of positive integers. Perform the following operation until grid becomes empty:

Delete the element with the greatest value from each row. If multiple such elements exist, delete any of them. Add the maximum of deleted elements to the answer.

Note that the number of columns decreases by one after each operation. Return the answer after performing the operations described above.   Example 1:

Input: grid = [[1,2,4],[3,3,1]] Output: 8 Explanation: The diagram above shows the removed values in each step. - In the first operation, we remove 4 from the first row and 3 from the second row (notice that, there are two cells with value 3 and we can remove any of them). We add 4 to the answer. - In the second operation, we remove 2 from the first row and 3 from the second row. We add 3 to the answer. - In the third operation, we remove 1 from the first row and 1 from the second row. We add 1 to the answer. The final answer = 4 + 3 + 1 = 8.

Example 2:

Input: grid = [[10]] Output: 10 Explanation: The diagram above shows the removed values in each step. - In the first operation, we remove 10 from the first row. We add 10 to the answer. The final answer = 10.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 50 1 <= grid[i][j] <= 100
==
Problem Number: 476 URL: https://leetcode.com/problems/count-pairs-of-similar-strings Title: 2506. Count Pairs Of Similar Strings Problem Description: You are given a 0-indexed string array words. Two strings are similar if they consist of the same characters.

For example, "abca" and "cba" are similar since both consist of characters 'a', 'b', and 'c'. However, "abacba" and "bcfd" are not similar since they do not consist of the same characters.

Return the number of pairs (i, j) such that 0 <= i < j <= word.length - 1 and the two strings words[i] and words[j] are similar.   Example 1: Input: words = ["aba","aabb","abcd","bac","aabc"] Output: 2 Explanation: There are 2 pairs that satisfy the conditions: - i = 0 and j = 1 : both words[0] and words[1] only consist of characters 'a' and 'b'. - i = 3 and j = 4 : both words[3] and words[4] only consist of characters 'a', 'b', and 'c'.

Example 2: Input: words = ["aabb","ab","ba"] Output: 3 Explanation: There are 3 pairs that satisfy the conditions: - i = 0 and j = 1 : both words[0] and words[1] only consist of characters 'a' and 'b'. - i = 0 and j = 2 : both words[0] and words[2] only consist of characters 'a' and 'b'. - i = 1 and j = 2 : both words[1] and words[2] only consist of characters 'a' and 'b'.

Example 3: Input: words = ["nba","cba","dba"] Output: 0 Explanation: Since there does not exist any pair that satisfies the conditions, we return 0.   Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 100 words[i] consist of only lowercase English letters.

==
Problem Number: 477 URL: https://leetcode.com/problems/maximum-enemy-forts-that-can-be-captured Title: 2511. Maximum Enemy Forts That Can Be Captured Problem Description: You are given a 0-indexed integer array forts of length n representing the positions of several forts. forts[i] can be -1, 0, or 1 where:

-1 represents there is no fort at the ith position. 0 indicates there is an enemy fort at the ith position. 1 indicates the fort at the ith the position is under your command.

Now you have decided to move your army from one of your forts at position i to an empty position j such that:

0 <= i, j <= n - 1 The army travels over enemy forts only. Formally, for all k where min(i,j) < k < max(i,j), forts[k] == 0.

While moving the army, all the enemy forts that come in the way are captured. Return the maximum number of enemy forts that can be captured. In case it is impossible to move your army, or you do not have any fort under your command, return 0. Example 1: Input: forts = [1,0,0,-1,0,0,0,0,1] Output: 4 Explanation: - Moving the army from position 0 to position 3 captures 2 enemy forts, at 1 and 2. - Moving the army from position 8 to position 3 captures 4 enemy forts. Since 4 is the maximum number of enemy forts that can be captured, we return 4.

Example 2: Input: forts = [0,0,1,-1] Output: 0 Explanation: Since no enemy fort can be captured, 0 is returned.

Constraints:

1 <= forts.length <= 1000 -1 <= forts[i] <= 1
==
Problem Number: 478 URL: https://leetcode.com/problems/shortest-distance-to-target-string-in-a-circular-array Title: 2515. Shortest Distance to Target String in a Circular Array Problem Description: You are given a 0-indexed circular string array words and a string target. A circular array means that the array's end connects to the array's beginning.

Formally, the next element of words[i] is words[(i + 1) % n] and the previous element of words[i] is words[(i - 1 + n) % n], where n is the length of words.

Starting from startIndex, you can move to either the next word or the previous word with 1 step at a time. Return the shortest distance needed to reach the string target. If the string target does not exist in words, return -1. Example 1: Input: words = ["hello","i","am","leetcode","hello"], target = "hello", startIndex = 1 Output: 1 Explanation: We start from index 1 and can reach "hello" by - moving 3 units to the right to reach index 4. - moving 2 units to the left to reach index 4. - moving 4 units to the right to reach index 0. - moving 1 unit to the left to reach index 0. The shortest distance to reach "hello" is 1.

Example 2: Input: words = ["a","b","leetcode"], target = "leetcode", startIndex = 0 Output: 1 Explanation: We start from index 0 and can reach "leetcode" by - moving 2 units to the right to reach index 3. - moving 1 unit to the left to reach index 3. The shortest distance to reach "leetcode" is 1. Example 3: Input: words = ["i","eat","leetcode"], target = "ate", startIndex = 0 Output: -1 Explanation: Since "ate" does not exist in words, we return -1.

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 100 words[i] and target consist of only lowercase English letters. 0 <= startIndex < words.length

==
Problem Number: 479 URL: https://leetcode.com/problems/count-the-digits-that-divide-a-number Title: 2520. Count the Digits That Divide a Number Problem Description: Given an integer num, return the number of digits in num that divide num. An integer val divides nums if nums % val == 0. Example 1: Input: num = 7 Output: 1 Explanation: 7 divides itself, hence the answer is 1.

Example 2: Input: num = 121 Output: 2 Explanation: 121 is divisible by 1, but not 2. Since 1 occurs twice as a digit, we return 2.

Example 3: Input: num = 1248 Output: 4 Explanation: 1248 is divisible by all of its digits, hence the answer is 4.

Constraints:

1 <= num <= 109 num does not contain 0 as one of its digits.

==
Problem Number: 480 URL: https://leetcode.com/problems/categorize-box-according-to-criteria Title: 2525. Categorize Box According to Criteria Problem Description: Given four integers length, width, height, and mass, representing the dimensions and mass of a box, respectively, return a string representing the category of the box.

The box is "Bulky" if:

Any of the dimensions of the box is greater or equal to 104. Or, the volume of the box is greater or equal to 109.

If the mass of the box is greater or equal to 100, it is "Heavy". If the box is both "Bulky" and "Heavy", then its category is "Both". If the box is neither "Bulky" nor "Heavy", then its category is "Neither". If the box is "Bulky" but not "Heavy", then its category is "Bulky". If the box is "Heavy" but not "Bulky", then its category is "Heavy".

Note that the volume of the box is the product of its length, width and height. Example 1: Input: length = 1000, width = 35, height = 700, mass = 300 Output: "Heavy" Explanation: None of the dimensions of the box is greater or equal to 104. Its volume = 24500000 <= 109. So it cannot be categorized as "Bulky". However mass >= 100, so the box is "Heavy". Since the box is not "Bulky" but "Heavy", we return "Heavy". Example 2: Input: length = 200, width = 50, height = 800, mass = 50 Output: "Neither" Explanation: None of the dimensions of the box is greater or equal to 104. Its volume = 8 * 106 <= 109. So it cannot be categorized as "Bulky". Its mass is also less than 100, so it cannot be categorized as "Heavy" either. Since its neither of the two above categories, we return "Neither". Constraints:

1 <= length, width, height <= 105 1 <= mass <= 103

==
Problem Number: 481 URL: https://leetcode.com/problems/maximum-count-of-positive-integer-and-negative-integer Title: 2529. Maximum Count of Positive Integer and Negative Integer Problem Description: Given an array nums sorted in non-decreasing order, return the maximum between the number of positive integers and the number of negative integers.

In other words, if the number of positive integers in nums is pos and the number of negative integers is neg, then return the maximum of pos and neg.

Note that 0 is neither positive nor negative. Example 1: Input: nums = [-2,-1,-1,1,2,3] Output: 3 Explanation: There are 3 positive integers and 3 negative integers. The maximum count among them is 3.

Example 2: Input: nums = [-3,-2,-1,0,0,1,2] Output: 3 Explanation: There are 2 positive integers and 3 negative integers. The maximum count among them is 3.

Example 3: Input: nums = [5,20,66,1314] Output: 4 Explanation: There are 4 positive integers and 0 negative integers. The maximum count among them is 4.

Constraints:

1 <= nums.length <= 2000 -2000 <= nums[i] <= 2000 nums is sorted in a non-decreasing order.

Follow up: Can you solve the problem in O(log(n)) time complexity?

==
Problem Number: 482 URL: https://leetcode.com/problems/difference-between-element-sum-and-digit-sum-of-an-array Title: 2535. Difference Between Element Sum and Digit Sum of an Array Problem Description: You are given a positive integer array nums.

The element sum is the sum of all the elements in nums. The digit sum is the sum of all the digits (not necessarily distinct) that appear in nums.

Return the absolute difference between the element sum and digit sum of nums. Note that the absolute difference between two integers x and y is defined as |x - y|. Example 1: Input: nums = [1,15,6,3] Output: 9 Explanation: The element sum of nums is 1 + 15 + 6 + 3 = 25. The digit sum of nums is 1 + 1 + 5 +

6 + 3 = 16. The absolute difference between the element sum and digit sum is |25 - 16| = 9.

Example 2: Input: nums = [1,2,3,4] Output: 0 Explanation: The element sum of nums is 1 + 2 + 3 + 4 = 10. The digit sum of nums is 1 + 2 + 3 + 4 = 10. The absolute difference between the element sum and digit sum is |10 - 10| = 0.

Constraints:

1 <= nums.length <= 2000 1 <= nums[i] <= 2000

==
Problem Number: 483 URL: https://leetcode.com/problems/minimum-common-value Title: 2540. Minimum Common Value Problem Description: Given two integer arrays nums1 and nums2, sorted in non-decreasing order, return the minimum integer common to both arrays. If there is no common integer amongst nums1 and nums2, return -1. Note that an integer is said to be common to nums1 and nums2 if both arrays have at least one occurrence of that integer. Example 1: Input: nums1 = [1,2,3], nums2 = [2,4] Output: 2 Explanation: The smallest element common to both arrays is 2, so we return 2.

Example 2: Input: nums1 = [1,2,3,6], nums2 = [2,3,4,5] Output: 2 Explanation: There are two common elements in the array 2 and 3 out of which 2 is the smallest, so 2 is returned.

Constraints:

1 <= nums1.length, nums2.length <= 105 1 <= nums1[i], nums2[j] <= 109 Both nums1 and nums2 are sorted in non-decreasing order.

==
Problem Number: 484 URL: https://leetcode.com/problems/alternating-digit-sum Title: 2544. Alternating Digit Sum Problem Description: You are given a positive integer n. Each digit of n has a sign according to the following rules:

The most significant digit is assigned a positive sign. Each other digit has an opposite sign to its adjacent digits.

Return the sum of all digits with their corresponding sign. Example 1: Input: n = 521 Output: 4 Explanation: (+5) + (-2) + (+1) = 4.

Example 2: Input: n = 111 Output: 1 Explanation: (+1) + (-1) + (+1) = 1.

Example 3: Input: n = 886996 Output: 0 Explanation: (+8) + (-8) + (+6) + (-9) + (+9) + (-6) = 0.

Constraints:

1 <= n <= 109

==
Problem Number: 485 URL: https://leetcode.com/problems/count-distinct-

numbers-on-board Title: 2549. Count Distinct Numbers on Board Problem Description: You are given a positive integer n, that is initially placed on a board. Every day, for 109 days, you perform the following procedure:

For each number x present on the board, find all numbers 1 <= i <= n such that x % i == 1. Then, place those numbers on the board.

Return the number of distinct integers present on the board after 109 days have elapsed. Note:

Once a number is placed on the board, it will remain on it until the end. % stands for the modulo operation. For example, 14 % 3 is 2.

Example 1: Input: n = 5 Output: 4 Explanation: Initially, 5 is present on the board. The next day, 2 and 4 will be added since 5 % 2 == 1 and 5 % 4 == 1. After that day, 3 will be added to the board because 4 % 3 == 1. At the end of a billion days, the distinct numbers on the board will be 2, 3, 4, and 5.

Example 2: Input: n = 3 Output: 2 Explanation: Since 3 % 2 == 1, 2 will be added to the board. After a billion days, the only two distinct numbers on the board are 2 and 3.

Constraints:

1 <= n <= 100

==

Problem Number: 486 URL: https://leetcode.com/problems/separate-the-digits-in-an-array Title: 2553. Separate the Digits in an Array Problem Description: Given an array of positive integers nums, return an array answer that consists of the digits of each integer in nums after separating them in the same order they appear in nums. To separate the digits of an integer is to get all the digits it has in the same order.

For example, for the integer 10921, the separation of its digits is [1,0,9,2,1].

Example 1: Input: nums = [13,25,83,77] Output: [1,3,2,5,8,3,7,7] Explanation: - The separation of 13 is [1,3]. - The separation of 25 is [2,5]. - The separation of 83 is [8,3]. - The separation of 77 is [7,7]. answer = [1,3,2,5,8,3,7,7]. Note that answer contains the separations in the same order.

Example 2: Input: nums = [7,1,3,9] Output: [7,1,3,9] Explanation: The separation of each integer in nums is itself. answer = [7,1,3,9].

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 105

==

Problem Number: 487 URL: https://leetcode.com/problems/take-gifts-from-the-richest-pile Title: 2558. Take Gifts From the Richest Pile Problem Description: You are given an integer array gifts denoting the number of gifts in various piles. Every second, you do the following:

Choose the pile with the maximum number of gifts. If there is more than one pile with the maximum number of gifts, choose any. Leave behind the floor of the square root of the number of gifts in the pile. Take the rest of the gifts.

Return the number of gifts remaining after k seconds. Example 1: Input: gifts = [25,64,9,4,100], k = 4 Output: 29 Explanation: The gifts are taken in the following way: - In the first second, the last pile is chosen and 10 gifts are left behind. - Then the second pile is chosen and 8 gifts are left behind. - After that the first pile is chosen and 5 gifts are left behind. - Finally, the last pile is chosen again and 3 gifts are left behind. The final remaining gifts are [5,8,9,4,3], so the total number of gifts remaining is 29.

Example 2: Input: gifts = [1,1,1,1], k = 4 Output: 4 Explanation: In this case, regardless which pile you choose, you have to leave behind 1 gift in each pile. That is, you can't take any pile with you. So, the total gifts remaining are 4.

Constraints:

1 <= gifts.length <= 103 1 <= gifts[i] <= 109 1 <= k <= 103

==

Problem Number: 488 URL: https://leetcode.com/problems/find-the-array-concatenation-value Title: 2562. Find the Array Concatenation Value Problem Description: You are given a 0-indexed integer array nums. The concatenation of two numbers is the number formed by concatenating their numerals.

For example, the concatenation of 15, 49 is 1549.

The concatenation value of nums is initially equal to 0. Perform this operation until nums becomes empty:

If there exists more than one number in nums, pick the first element and last element in nums respectively and add the value of their concatenation to the concatenation value of nums, then delete the first and last element from nums. If one element exists, add its value to the concatenation value of nums, then delete it.

Return the concatenation value of the nums. Example 1: Input: nums = [7,52,2,4] Output: 596 Explanation: Before performing any operation, nums is [7,52,2,4] and concatenation value is 0. - In the first operation: We pick the first element, 7, and the last element, 4. Their concatenation is 74, and we add it to the concatenation value, so it becomes equal to 74. Then we delete them from nums, so nums becomes equal to [52,2]. - In the second operation: We pick the first element, 52, and the last element, 2. Their concatenation is 522, and we add it to the concatenation value, so it becomes equal to 596. Then we delete them from the nums, so nums becomes empty. Since the concatenation value is 596 so the answer is 596.

Example 2: Input: nums = [5,14,13,8,12] Output: 673 Explanation: Before performing any operation, nums is [5,14,13,8,12] and concatenation value is 0. - In the first operation: We pick the first element, 5, and the last element, 12.

Their concatenation is 512, and we add it to the concatenation value, so it becomes equal to 512. Then we delete them from the nums, so nums becomes equal to [14,13,8]. - In the second operation: We pick the first element, 14, and the last element, 8. Their concatenation is 148, and we add it to the concatenation value, so it becomes equal to 660. Then we delete them from the nums, so nums becomes equal to [13]. - In the third operation: nums has only one element, so we pick 13 and add it to the concatenation value, so it becomes equal to 673. Then we delete it from nums, so nums become empty. Since the concatenation value is 673 so the answer is 673.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 104

==
Problem Number: 489 URL: https://leetcode.com/problems/maximum-difference-by-remapping-a-digit Title: 2566. Maximum Difference by Remapping a Digit Problem Description: You are given an integer num. You know that Danny Mittal will sneakily remap one of the 10 possible digits (0 to 9) to another digit. Return the difference between the maximum and minimum values Danny can make by remapping exactly one digit in num. Notes:

When Danny remaps a digit d1 to another digit d2, Danny replaces all occurrences of d1 in num with d2. Danny can remap a digit to itself, in which case num does not change. Danny can remap different digits for obtaining minimum and maximum values respectively. The resulting number after remapping can contain leading zeroes. We mentioned ”Danny Mittal” to congratulate him on being in the top 10 in Weekly Contest 326.

Example 1: Input: num = 11891 Output: 99009 Explanation: To achieve the maximum value, Danny can remap the digit 1 to the digit 9 to yield 99899. To achieve the minimum value, Danny can remap the digit 1 to the digit 0, yielding 890. The difference between these two numbers is 99009.

Example 2: Input: num = 90 Output: 99 Explanation: The maximum value that can be returned by the function is 99 (if 0 is replaced by 9) and the minimum value that can be returned by the function is 0 (if 9 is replaced by 0). Thus, we return 99. Constraints:

1 <= num <= 108

==
Problem Number: 490 URL: https://leetcode.com/problems/merge-two-2d-arrays-by-summing-values Title: 2570. Merge Two 2D Arrays by Summing Values Problem Description: You are given two 2D integer arrays nums1 and nums2.

nums1[i] = [idi, vali] indicate that the number with the id idi has a value equal to vali. nums2[i] = [idi, vali] indicate that the number with the id idi has a value equal to vali.

Each array contains unique ids and is sorted in ascending order by id. Merge the two arrays into one array that is sorted in ascending order by id, respecting the following conditions:

Only ids that appear in at least one of the two arrays should be included in the resulting array. Each id should be included only once and its value should be the sum of the values of this id in the two arrays. If the id does not exist in one of the two arrays then its value in that array is considered to be 0.

Return the resulting array. The returned array must be sorted in ascending order by id. Example 1: Input: nums1 = [[1,2],[2,3],[4,5]], nums2 = [[1,4],[3,2],[4,1]] Output: [[1,6],[2,3],[3,2],[4,6]] Explanation: The resulting array contains the following: - id = 1, the value of this id is 2 + 4 = 6. - id = 2, the value of this id is 3. - id = 3, the value of this id is 2. - id = 4, the value of this id is 5 + 1 = 6.

Example 2: Input: nums1 = [[2,4],[3,6],[5,5]], nums2 = [[1,3],[4,3]] Output: [[1,3],[2,4],[3,6],[4,3],[5,5]] Explanation: There are no common ids, so we just include each id with its value in the resulting list.

Constraints:

1 <= nums1.length, nums2.length <= 200 nums1[i].length == nums2[j].length == 2 1 <= idi, vali <= 1000 Both arrays contain unique ids. Both arrays are in strictly ascending order by id.

==
Problem Number: 491 URL: https://leetcode.com/problems/left-and-right-sum-differences Title: 2574. Left and Right Sum Differences Problem Description: Given a 0-indexed integer array nums, find a 0-indexed integer array answer where:

answer.length == nums.length. answer[i] = |leftSum[i] - rightSum[i]|.

Where:

leftSum[i] is the sum of elements to the left of the index i in the array nums. If there is no such element, leftSum[i] = 0. rightSum[i] is the sum of elements to the right of the index i in the array nums. If there is no such element, rightSum[i] = 0.

Return the array answer. Example 1: Input: nums = [10,4,8,3] Output: [15,1,11,22] Explanation: The array leftSum is [0,10,14,22] and the array rightSum is [15,11,3,0]. The array answer is [|0 - 15|,|10 - 11|,|14 - 3|,|22 - 0|] = [15,1,11,22].

Example 2: Input: nums = [1] Output: [0] Explanation: The array leftSum is [0] and the array rightSum is [0]. The array answer is [|0 - 0|] = [0].

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 105

==

Problem Number: 492 URL: https://leetcode.com/problems/split-with-minimum-sum Title: 2578. Split With Minimum Sum Problem Description: Given a positive integer num, split it into two non-negative integers num1 and num2 such that:

The concatenation of num1 and num2 is a permutation of num.

In other words, the sum of the number of occurrences of each digit in num1 and num2 is equal to the number of occurrences of that digit in num.

num1 and num2 can contain leading zeros.

Return the minimum possible sum of num1 and num2. Notes:

It is guaranteed that num does not contain any leading zeros. The order of occurrence of the digits in num1 and num2 may differ from the order of occurrence of num.

Example 1: Input: num = 4325 Output: 59 Explanation: We can split 4325 so that num1 is 24 and num2 is 35, giving a sum of 59. We can prove that 59 is indeed the minimal possible sum.

Example 2: Input: num = 687 Output: 75 Explanation: We can split 687 so that num1 is 68 and num2 is 7, which would give an optimal sum of 75.

Constraints:

10 <= num <= 109

==

Problem Number: 493 URL: https://leetcode.com/problems/pass-the-pillow Title: 2582. Pass the Pillow Problem Description: There are n people standing in a line labeled from 1 to n. The first person in the line is holding a pillow initially. Every second, the person holding the pillow passes it to the next person standing in the line. Once the pillow reaches the end of the line, the direction changes, and people continue passing the pillow in the opposite direction.

For example, once the pillow reaches the nth person they pass it to the n - 1th person, then to the n - 2th person and so on.

Given the two positive integers n and time, return the index of the person holding the pillow after time seconds. Example 1: Input: n = 4, time = 5 Output: 2 Explanation: People pass the pillow in the following way: 1 -> 2 -> 3 -> 4 -> 3 -> 2. Afer five seconds, the pillow is given to the 2nd person.

Example 2: Input: n = 3, time = 2 Output: 3 Explanation: People pass the pillow in the following way: 1 -> 2 -> 3. Afer two seconds, the pillow is given to the 3rd person.

Constraints:

2 <= n <= 1000 1 <= time <= 1000

==
Problem Number: 494 URL: https://leetcode.com/problems/count-the-number-of-vowel-strings-in-range Title: 2586. Count the Number of Vowel Strings in Range Problem Description: You are given a 0-indexed array of string words and two integers left and right. A string is called a vowel string if it starts with a vowel character and ends with a vowel character where vowel characters are 'a', 'e', 'i', 'o', and 'u'. Return the number of vowel strings words[i] where i belongs to the inclusive range [left, right].   Example 1: Input: words = ["are","amy","u"], left = 0, right = 2 Output: 2 Explanation: - "are" is a vowel string because it starts with 'a' and ends with 'e'. - "amy" is not a vowel string because it does not end with a vowel. - "u" is a vowel string because it starts with 'u' and ends with 'u'. The number of vowel strings in the mentioned range is 2.

Example 2: Input: words = ["hey","aeo","mu","ooo","artro"], left = 1, right = 4 Output: 3 Explanation: - "aeo" is a vowel string because it starts with 'a' and ends with 'o'. - "mu" is not a vowel string because it does not start with a vowel. - "ooo" is a vowel string because it starts with 'o' and ends with 'o'. - "artro" is a vowel string because it starts with 'a' and ends with 'o'. The number of vowel strings in the mentioned range is 3.

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 10 words[i] consists of only lowercase English letters. 0 <= left <= right < words.length

==
Problem Number: 495 URL: https://leetcode.com/problems/distribute-money-to-maximum-children Title: 2591. Distribute Money to Maximum Children Problem Description: You are given an integer money denoting the amount of money (in dollars) that you have and another integer children denoting the number of children that you must distribute the money to. You have to distribute the money according to the following rules:

All money must be distributed. Everyone must receive at least 1 dollar. Nobody receives 4 dollars.

Return the maximum number of children who may receive exactly 8 dollars if you distribute the money according to the aforementioned rules. If there is no way to distribute the money, return -1.   Example 1: Input: money = 20, children = 3 Output: 1 Explanation: The maximum number of children with 8 dollars will be 1. One of the ways to distribute the money is: - 8 dollars to the first child. - 9 dollars to the second child. - 3 dollars to the third child. It can be proven that no distribution exists such that number of children getting 8 dollars is greater than 1.

Example 2: Input: money = 16, children = 2 Output: 2 Explanation: Each child can be given 8 dollars.

Constraints:

1 <= money <= 200 2 <= children <= 30

==

Problem Number: 496 URL: https://leetcode.com/problems/number-of-even-and-odd-bits Title: 2595. Number of Even and Odd Bits Problem Description: You are given a positive integer n. Let even denote the number of even indices in the binary representation of n (0-indexed) with value 1. Let odd denote the number of odd indices in the binary representation of n (0-indexed) with value 1. Return an integer array answer where answer = [even, odd]. Example 1: Input: n = 17 Output: [2,0] Explanation: The binary representation of 17 is 10001. It contains 1 on the 0th and 4th indices. There are 2 even and 0 odd indices.

Example 2: Input: n = 2 Output: [0,1] Explanation: The binary representation of 2 is 10. It contains 1 on the 1st index. There are 0 even and 1 odd indices.

Constraints:

1 <= n <= 1000

==

Problem Number: 497 URL: https://leetcode.com/problems/k-items-with-the-maximum-sum Title: 2600. K Items With the Maximum Sum Problem Description: There is a bag that consists of items, each item has a number 1, 0, or -1 written on it. You are given four non-negative integers numOnes, numZeros, numNegOnes, and k. The bag initially contains:

numOnes items with 1s written on them. numZeroes items with 0s written on them. numNegOnes items with -1s written on them.

We want to pick exactly k items among the available items. Return the maximum possible sum of numbers written on the items. Example 1: Input: numOnes = 3, numZeros = 2, numNegOnes = 0, k = 2 Output: 2 Explanation: We have a bag of items with numbers written on them {1, 1, 1, 0, 0}. We take 2 items with 1 written on them and get a sum in a total of 2. It can be proven that 2 is the maximum possible sum.

Example 2: Input: numOnes = 3, numZeros = 2, numNegOnes = 0, k = 4 Output: 3 Explanation: We have a bag of items with numbers written on them {1, 1, 1, 0, 0}. We take 3 items with 1 written on them, and 1 item with 0 written on it, and get a sum in a total of 3. It can be proven that 3 is the maximum possible sum.

Constraints:

0 <= numOnes, numZeros, numNegOnes <= 50 0 <= k <= numOnes + numZeros + numNegOnes

==

Problem Number: 498 URL: https://leetcode.com/problems/form-smallest-number-from-two-digit-arrays Title: 2605. Form Smallest Number From Two Digit Arrays Problem Description: Given two arrays of unique digits nums1 and nums2, return the smallest number that contains at least one digit from each array. Example 1: Input: nums1 = [4,1,3], nums2 = [5,7] Output: 15 Explanation: The number 15 contains the digit 1 from nums1 and the digit 5 from nums2. It can be proven that 15 is the smallest number we can have.

Example 2: Input: nums1 = [3,5,2,6], nums2 = [3,1,7] Output: 3 Explanation: The number 3 contains the digit 3 which exists in both arrays.

Constraints:

1 <= nums1.length, nums2.length <= 9 1 <= nums1[i], nums2[i] <= 9 All digits in each array are unique.

==

Problem Number: 499 URL: https://leetcode.com/problems/find-the-longest-balanced-substring-of-a-binary-string Title: 2609. Find the Longest Balanced Substring of a Binary String Problem Description: You are given a binary string s consisting only of zeroes and ones. A substring of s is considered balanced if all zeroes are before ones and the number of zeroes is equal to the number of ones inside the substring. Notice that the empty substring is considered a balanced substring. Return the length of the longest balanced substring of s. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "01000111" Output: 6 Explanation: The longest balanced substring is "000111", which has length 6.

Example 2: Input: s = "00111" Output: 4 Explanation: The longest balanced substring is "0011", which has length 4.

Example 3: Input: s = "111" Output: 0 Explanation: There is no balanced substring except the empty substring, so the answer is 0.

Constraints:

1 <= s.length <= 50 '0' <= s[i] <= '1'

==

Problem Number: 500 URL: https://leetcode.com/problems/prime-in-diagonal Title: 2614. Prime In Diagonal Problem Description: You are given a 0-indexed two-dimensional integer array nums. Return the largest prime number that lies on at least one of the diagonals of nums. In case, no prime is present on any of the diagonals, return 0. Note that:

An integer is prime if it is greater than 1 and has no positive integer divisors other than 1 and itself. An integer val is on one of the diagonals of nums if there exists an integer i for which nums[i][i] = val or an i for which nums[i][nums.length - i - 1] = val.

In the above diagram, one diagonal is [1,5,9] and another diagonal is [3,5,7]. Example 1: Input: nums = [[1,2,3],[5,6,7],[9,10,11]] Output: 11 Explanation: The numbers 1, 3, 6, 9, and 11 are the only numbers present on at least one of the diagonals. Since 11 is the largest prime, we return 11.

Example 2: Input: nums = [[1,2,3],[5,17,7],[9,11,10]] Output: 17 Explanation: The numbers 1, 3, 9, 10, and 17 are all present on at least one of the diagonals. 17 is the largest prime, so we return 17.

Constraints:

1 <= nums.length <= 300 nums.length == numsi.length 1 <= nums[i][j] <= 4*106

== Problem Number: 501 URL: https://leetcode.com/problems/find-the-width-of-columns-of-a-grid Title: 2639. Find the Width of Columns of a Grid Problem Description: You are given a 0-indexed m x n integer matrix grid. The width of a column is the maximum length of its integers.

For example, if grid = [[-10], [3], [12]], the width of the only column is 3 since -10 is of length 3.

Return an integer array ans of size n where ans[i] is the width of the ith column. The length of an integer x with len digits is equal to len if x is non-negative, and len + 1 otherwise. Example 1: Input: grid = [[1],[22],[333]] Output: [3] Explanation: In the 0th column, 333 is of length 3.

Example 2: Input: grid = [[-15,1,3],[15,7,12],[5,6,-2]] Output: [3,1,2] Explanation: In the 0th column, only -15 is of length 3. In the 1st column, all integers are of length 1. In the 2nd column, both 12 and -2 are of length 2.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 100 -109 <= grid[r][c] <= 109

== Problem Number: 502 URL: https://leetcode.com/problems/row-with-maximum-ones Title: 2643. Row With Maximum Ones Problem Description: Given a m x n binary matrix mat, find the 0-indexed position of the row that contains the maximum count of ones, and the number of ones in that row. In case there are multiple rows that have the maximum count of ones, the row with the smallest row number should be selected. Return an array containing the index of the row, and the number of ones in it. Example 1: Input: mat = [[0,1],[1,0]] Output: [0,1] Explanation: Both rows have the same number of 1's. So we return the index of the smaller row, 0, and the maximum count of ones (1). So, the answer is [0,1].

Example 2: Input: mat = [[0,0,0],[0,1,1]] Output: [1,2] Explanation: The row indexed 1 has the maximum count of ones (2). So we return its index, 1, and the count. So, the answer is [1,2].

Example 3: Input: mat = [[0,0],[1,1],[0,0]] Output: [1,2] Explanation: The row indexed 1 has the maximum count of ones (2). So the answer is [1,2].

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 100 mat[i][j] is either 0 or 1.

==
Problem Number: 503 URL: https://leetcode.com/problems/find-the-maximum-divisibility-score Title: 2644. Find the Maximum Divisibility Score Problem Description: You are given two 0-indexed integer arrays nums and divisors. The divisibility score of divisors[i] is the number of indices j such that nums[j] is divisible by divisors[i]. Return the integer divisors[i] with the maximum divisibility score. If there is more than one integer with the maximum score, return the minimum of them. Example 1: Input: nums = [4,7,9,3,9], divisors = [5,2,3] Output: 3 Explanation: The divisibility score for every element in divisors is: The divisibility score of divisors[0] is 0 since no number in nums is divisible by 5. The divisibility score of divisors[1] is 1 since nums[0] is divisible by 2. The divisibility score of divisors[2] is 3 since nums[2], nums[3], and nums[4] are divisible by 3. Since divisors[2] has the maximum divisibility score, we return it.

Example 2: Input: nums = [20,14,21,10], divisors = [5,7,5] Output: 5 Explanation: The divisibility score for every element in divisors is: The divisibility score of divisors[0] is 2 since nums[0] and nums[3] are divisible by 5. The divisibility score of divisors[1] is 2 since nums[1] and nums[2] are divisible by 7. The divisibility score of divisors[2] is 2 since nums[0] and nums[3] are divisible by 5. Since divisors[0], divisors[1], and divisors[2] all have the maximum divisibility score, we return the minimum of them (i.e., divisors[2]).

Example 3: Input: nums = [12], divisors = [10,16] Output: 10 Explanation: The divisibility score for every element in divisors is: The divisibility score of divisors[0] is 0 since no number in nums is divisible by 10. The divisibility score of divisors[1] is 0 since no number in nums is divisible by 16. Since divisors[0] and divisors[1] both have the maximum divisibility score, we return the minimum of them (i.e., divisors[0]).

Constraints:

1 <= nums.length, divisors.length <= 1000 1 <= nums[i], divisors[i] <= 109

==
Problem Number: 504 URL: https://leetcode.com/problems/calculate-delayed-arrival-time Title: 2651. Calculate Delayed Arrival Time Problem Description: You are given a positive integer arrivalTime denoting the arrival time of a

train in hours, and another positive integer delayedTime denoting the amount of delay in hours. Return the time when the train will arrive at the station. Note that the time in this problem is in 24-hours format. Example 1: Input: arrivalTime = 15, delayedTime = 5 Output: 20 Explanation: Arrival time of the train was 15:00 hours. It is delayed by 5 hours. Now it will reach at 15+5 = 20 (20:00 hours).

Example 2: Input: arrivalTime = 13, delayedTime = 11 Output: 0 Explanation: Arrival time of the train was 13:00 hours. It is delayed by 11 hours. Now it will reach at 13+11=24 (Which is denoted by 00:00 in 24 hours format so return 0).

Constraints:

1 <= arrivaltime < 24 1 <= delayedTime <= 24

==
Problem Number: 505 URL: https://leetcode.com/problems/sum-multiples Title: 2652. Sum Multiples Problem Description: Given a positive integer n, find the sum of all integers in the range [1, n] inclusive that are divisible by 3, 5, or 7. Return an integer denoting the sum of all numbers in the given range satisfying the constraint. Example 1: Input: n = 7 Output: 21 Explanation: Numbers in the range [1, 7] that are divisible by 3, 5, or 7 are 3, 5, 6, 7. The sum of these numbers is 21.

Example 2: Input: n = 10 Output: 40 Explanation: Numbers in the range [1, 10] that are divisible by 3, 5, or 7 are 3, 5, 6, 7, 9, 10. The sum of these numbers is 40.

Example 3: Input: n = 9 Output: 30 Explanation: Numbers in the range [1, 9] that are divisible by 3, 5, or 7 are 3, 5, 6, 7, 9. The sum of these numbers is 30.

Constraints:

1 <= n <= 103

==
Problem Number: 506 URL: https://leetcode.com/problems/maximum-sum-with-exactly-k-elements Title: 2656. Maximum Sum With Exactly K Elements Problem Description: You are given a 0-indexed integer array nums and an integer k. Your task is to perform the following operation exactly k times in order to maximize your score:

Select an element m from nums. Remove the selected element m from the array. Add a new element with a value of m + 1 to the array. Increase your score by m.

Return the maximum score you can achieve after performing the operation exactly k times. Example 1: Input: nums = [1,2,3,4,5], k = 3 Output: 18 Explanation: We need to choose exactly 3 elements from nums to maximize the sum. For the first iteration, we choose 5. Then sum is 5 and nums = [1,2,3,4,6] For the second iteration, we choose 6. Then sum is 5 + 6 and nums = [1,2,3,4,7]

For the third iteration, we choose 7. Then sum is 5 + 6 + 7 = 18 and nums = [1,2,3,4,8] So, we will return 18. It can be proven, that 18 is the maximum answer that we can achieve.

Example 2: Input: nums = [5,5,5], k = 2 Output: 11 Explanation: We need to choose exactly 2 elements from nums to maximize the sum. For the first iteration, we choose 5. Then sum is 5 and nums = [5,5,6] For the second iteration, we choose 6. Then sum is 5 + 6 = 11 and nums = [5,5,7] So, we will return 11. It can be proven, that 11 is the maximum answer that we can achieve.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100 1 <= k <= 100

==
Problem Number: 507 URL: https://leetcode.com/problems/determine-the-winner-of-a-bowling-game Title: 2660. Determine the Winner of a Bowling Game Problem Description: You are given two 0-indexed integer arrays player1 and player2, that represent the number of pins that player 1 and player 2 hit in a bowling game, respectively. The bowling game consists of n turns, and the number of pins in each turn is exactly 10. Assume a player hit xi pins in the ith turn. The value of the ith turn for the player is:

2xi if the player hit 10 pins in any of the previous two turns. Otherwise, It is xi.

The score of the player is the sum of the values of their n turns. Return

1 if the score of player 1 is more than the score of player 2, 2 if the score of player 2 is more than the score of player 1, and 0 in case of a draw.

Example 1: Input: player1 = [4,10,7,9], player2 = [6,5,2,3] Output: 1 Explanation: The score of player1 is 4 + 10 + 2*7 + 2*9 = 46. The score of player2 is 6 + 5 + 2 + 3 = 16. Score of player1 is more than the score of player2, so, player1 is the winner, and the answer is 1.

Example 2: Input: player1 = [3,5,7,6], player2 = [8,10,10,2] Output: 2 Explanation: The score of player1 is 3 + 5 + 7 + 6 = 21. The score of player2 is 8 + 10 + 2*10 + 2*2 = 42. Score of player2 is more than the score of player1, so, player2 is the winner, and the answer is 2.

Example 3: Input: player1 = [2,3], player2 = [4,1] Output: 0 Explanation: The score of player1 is 2 + 3 = 5 The score of player2 is 4 + 1 = 5 The score of player1 equals to the score of player2, so, there is a draw, and the answer is 0.

Constraints:

n == player1.length == player2.length 1 <= n <= 1000 0 <= player1[i], player2[i] <= 10

==
Problem Number: 508 URL: https://leetcode.com/problems/find-the-distinct-difference-array Title: 2670. Find the Distinct Difference Array Problem Description: You are given a 0-indexed array nums of length n. The distinct difference array of nums is an array diff of length n such that diff[i] is equal to the number of distinct elements in the suffix nums[i + 1, ..., n - 1] subtracted from the number of distinct elements in the prefix nums[0, ..., i]. Return the distinct difference array of nums. Note that nums[i, ..., j] denotes the subarray of nums starting at index i and ending at index j inclusive. Particularly, if i > j then nums[i, ..., j] denotes an empty subarray. Example 1: Input: nums = [1,2,3,4,5] Output: [-3,-1,1,3,5] Explanation: For index i = 0, there is 1 element in the prefix and 4 distinct elements in the suffix. Thus, diff[0] = 1 - 4 = -3. For index i = 1, there are 2 distinct elements in the prefix and 3 distinct elements in the suffix. Thus, diff[1] = 2 - 3 = -1. For index i = 2, there are 3 distinct elements in the prefix and 2 distinct elements in the suffix. Thus, diff[2] = 3 - 2 = 1. For index i = 3, there are 4 distinct elements in the prefix and 1 distinct element in the suffix. Thus, diff[3] = 4 - 1 = 3. For index i = 4, there are 5 distinct elements in the prefix and no elements in the suffix. Thus, diff[4] = 5 - 0 = 5.

Example 2: Input: nums = [3,2,3,4,2] Output: [-2,-1,0,2,3] Explanation: For index i = 0, there is 1 element in the prefix and 3 distinct elements in the suffix. Thus, diff[0] = 1 - 3 = -2. For index i = 1, there are 2 distinct elements in the prefix and 3 distinct elements in the suffix. Thus, diff[1] = 2 - 3 = -1. For index i = 2, there are 2 distinct elements in the prefix and 2 distinct elements in the suffix. Thus, diff[2] = 2 - 2 = 0. For index i = 3, there are 3 distinct elements in the prefix and 1 distinct element in the suffix. Thus, diff[3] = 3 - 1 = 2. For index i = 4, there are 3 distinct elements in the prefix and no elements in the suffix. Thus, diff[4] = 3 - 0 = 3.

Constraints:

1 <= n == nums.length <= 50 1 <= nums[i] <= 50

==
Problem Number: 509 URL: https://leetcode.com/problems/number-of-senior-citizens Title: 2678. Number of Senior Citizens Problem Description: You are given a 0-indexed array of strings details. Each element of details provides information about a given passenger compressed into a string of length 15. The system is such that:

The first ten characters consist of the phone number of passengers. The next character denotes the gender of the person. The following two characters are used to indicate the age of the person. The last two characters determine the seat allotted to that person.

Return the number of passengers who are strictly more than 60 years old. Example 1: Input: details = ["7868190130M7522","5303914400F9211","9273338290F4010"]

Output: 2 Explanation: The passengers at indices 0, 1, and 2 have ages 75, 92, and 40. Thus, there are 2 people who are over 60 years old.

Example 2: Input: details = [”1313579440F2036”,”2921522980M5644”] Output: 0 Explanation: None of the passengers are older than 60.

Constraints:

1 <= details.length <= 100 details[i].length == 15 details[i] consists of digits from ’0’ to ’9’. details[i][10] is either ’M’ or ’F’ or ’O’. The phone numbers and seat numbers of the passengers are distinct.

==
Problem Number: 510 URL: https://leetcode.com/problems/find-the-losers-of-the-circular-game Title: 2682. Find the Losers of the Circular Game Problem Description: There are n friends that are playing a game. The friends are sitting in a circle and are numbered from 1 to n in clockwise order. More formally, moving clockwise from the ith friend brings you to the (i+1)th friend for 1 <= i < n, and moving clockwise from the nth friend brings you to the 1st friend. The rules of the game are as follows: 1st friend receives the ball.

After that, 1st friend passes it to the friend who is k steps away from them in the clockwise direction. After that, the friend who receives the ball should pass it to the friend who is 2 * k steps away from them in the clockwise direction. After that, the friend who receives the ball should pass it to the friend who is 3 * k steps away from them in the clockwise direction, and so on and so forth.

In other words, on the ith turn, the friend holding the ball should pass it to the friend who is i * k steps away from them in the clockwise direction. The game is finished when some friend receives the ball for the second time. The losers of the game are friends who did not receive the ball in the entire game. Given the number of friends, n, and an integer k, return the array answer, which contains the losers of the game in the ascending order. Example 1: Input: n = 5, k = 2 Output: [4,5] Explanation: The game goes as follows: 1) Start at 1st friend and pass the ball to the friend who is 2 steps away from them - 3rd friend. 2) 3rd friend passes the ball to the friend who is 4 steps away from them - 2nd friend. 3) 2nd friend passes the ball to the friend who is 6 steps away from them - 3rd friend. 4) The game ends as 3rd friend receives the ball for the second time.

Example 2: Input: n = 4, k = 4 Output: [2,3,4] Explanation: The game goes as follows: 1) Start at the 1st friend and pass the ball to the friend who is 4 steps away from them - 1st friend. 2) The game ends as 1st friend receives the ball for the second time.

Constraints:

1 <= k <= n <= 50

==
Problem Number: 511 URL: https://leetcode.com/problems/minimum-string-

length-after-removing-substrings Title: 2696. Minimum String Length After Removing Substrings Problem Description: You are given a string s consisting only of uppercase English letters. You can apply some operations to this string where, in one operation, you can remove any occurrence of one of the substrings "AB" or "CD" from s. Return the minimum possible length of the resulting string that you can obtain. Note that the string concatenates after removing the substring and could produce new "AB" or "CD" substrings. Example 1: Input: s = "ABFCACDB" Output: 2 Explanation: We can do the following operations: - Remove the substring "ABFCACDB", so s = "FCACDB". - Remove the substring "FCACDB", so s = "FCAB". - Remove the substring "FCAB", so s = "FC". So the resulting length of the string is 2. It can be shown that it is the minimum length that we can obtain. Example 2: Input: s = "ACBBD" Output: 5 Explanation: We cannot do any operations on the string so the length remains the same.

Constraints:

1 <= s.length <= 100 s consists only of uppercase English letters.

==
Problem Number: 512 URL: https://leetcode.com/problems/lexicographically-smallest-palindrome Title: 2697. Lexicographically Smallest Palindrome Problem Description: You are given a string s consisting of lowercase English letters, and you are allowed to perform operations on it. In one operation, you can replace a character in s with another lowercase English letter. Your task is to make s a palindrome with the minimum number of operations possible. If there are multiple palindromes that can be made using the minimum number of operations, make the lexicographically smallest one. A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b. Return the resulting palindrome string. Example 1: Input: s = "egcfe" Output: "efcfe" Explanation: The minimum number of operations to make "egcfe" a palindrome is 1, and the lexicographically smallest palindrome string we can get by modifying one character is "efcfe", by changing 'g'.

Example 2: Input: s = "abcd" Output: "abba" Explanation: The minimum number of operations to make "abcd" a palindrome is 2, and the lexicographically smallest palindrome string we can get by modifying two characters is "abba".

Example 3: Input: s = "seven" Output: "neven" Explanation: The minimum number of operations to make "seven" a palindrome is 1, and the lexicographically smallest palindrome string we can get by modifying one character is "neven".

Constraints:

1 <= s.length <= 1000 s consists of only lowercase English letters.

==
Problem Number: 513 URL: https://leetcode.com/problems/buy-two-chocolates Title: 2706. Buy Two Chocolates Problem Description: You are given an integer array prices representing the prices of various chocolates in a store. You are also given a single integer money, which represents your initial amount of money. You must buy exactly two chocolates in such a way that you still have some non-negative leftover money. You would like to minimize the sum of the prices of the two chocolates you buy. Return the amount of money you will have leftover after buying the two chocolates. If there is no way for you to buy two chocolates without ending up in debt, return money. Note that the leftover must be non-negative. Example 1: Input: prices = [1,2,2], money = 3 Output: 0 Explanation: Purchase the chocolates priced at 1 and 2 units respectively. You will have 3 - 3 = 0 units of money afterwards. Thus, we return 0.

Example 2: Input: prices = [3,2,3], money = 3 Output: 3 Explanation: You cannot buy 2 chocolates without going in debt, so we return 3.

Constraints:

2 <= prices.length <= 50 1 <= prices[i] <= 100 1 <= money <= 100

==
Problem Number: 514 URL: https://leetcode.com/problems/remove-trailing-zeros-from-a-string Title: 2710. Remove Trailing Zeros From a String Problem Description: Given a positive integer num represented as a string, return the integer num without trailing zeros as a string. Example 1: Input: num = "51230100" Output: "512301" Explanation: Integer "51230100" has 2 trailing zeros, we remove them and return integer "512301".

Example 2: Input: num = "123" Output: "123" Explanation: Integer "123" has no trailing zeros, we return integer "123".

Constraints:

1 <= num.length <= 1000 num consists of only digits. num doesn't have any leading zeros.

==
Problem Number: 515 URL: https://leetcode.com/problems/minimize-string-length Title: 2716. Minimize String Length Problem Description: Given a 0-indexed string s, repeatedly perform the following operation any number of times:

Choose an index i in the string, and let c be the character in position i. Delete the closest occurrence of c to the left of i (if any) and the closest occurrence of c to the right of i (if any).

Your task is to minimize the length of s by performing the above operation any number of times. Return an integer denoting the length of the minimized string. Example 1: Input: s = "aaabc" Output: 3 Explanation: In this example, s is "aaabc". We can start by selecting the character 'a' at index 1. We then remove the closest 'a' to the left of index 1, which is at index 0, and the closest 'a' to the right of index 1, which is at index 2. After this operation, the string becomes "abc". Any further operation we perform on the string will leave it unchanged. Therefore, the length of the minimized string is 3. Example 2: Input: s = "cbbd" Output: 3 Explanation: For this we can start with character 'b' at index 1. There is no occurrence of 'b' to the left of index 1, but there is one to the right at index 2, so we delete the 'b' at index 2. The string becomes "cbd" and further operations will leave it unchanged. Hence, the minimized length is 3.

Example 3: Input: s = "dddaaa" Output: 2 Explanation: For this, we can start with the character 'd' at index 1. The closest occurrence of a 'd' to its left is at index 0, and the closest occurrence of a 'd' to its right is at index 2. We delete both index 0 and 2, so the string becomes "daaa". In the new string, we can select the character 'a' at index 2. The closest occurrence of an 'a' to its left is at index 1, and the closest occurrence of an 'a' to its right is at index 3. We delete both of them, and the string becomes "da". We cannot minimize this further, so the minimized length is 2.

Constraints:

1 <= s.length <= 100 s contains only lowercase English letters

==

Problem Number: 516 URL: https://leetcode.com/problems/semi-ordered-permutation Title: 2717. Semi-Ordered Permutation Problem Description: You are given a 0-indexed permutation of n integers nums. A permutation is called semi-ordered if the first number equals 1 and the last number equals n. You can perform the below operation as many times as you want until you make nums a semi-ordered permutation:

Pick two adjacent elements in nums, then swap them.

Return the minimum number of operations to make nums a semi-ordered permutation. A permutation is a sequence of integers from 1 to n of length n containing each number exactly once. Example 1: Input: nums = [2,1,4,3] Output: 2 Explanation: We can make the permutation semi-ordered using these sequence of operations: 1 - swap i = 0 and j = 1. The permutation becomes [1,2,4,3]. 2 - swap i = 2 and j = 3. The permutation becomes [1,2,3,4]. It can be proved that there is no sequence of less than two operations that make nums a semi-ordered permutation.

Example 2: Input: nums = [2,4,1,3] Output: 3 Explanation: We can make the permutation semi-ordered using these sequence of operations: 1 - swap i = 1 and j = 2. The permutation becomes [2,1,4,3]. 2 - swap i = 0 and j = 1. The permutation becomes [1,2,4,3]. 3 - swap i = 2 and j = 3. The permutation becomes [1,2,3,4]. It can be proved that there is no sequence of less than three operations that make nums a semi-ordered permutation.

Example 3: Input: nums = [1,3,4,2,5] Output: 0 Explanation: The permutation is already a semi-ordered permutation.

Constraints:

2 <= nums.length == n <= 50 1 <= nums[i] <= 50 nums is a permutation.

==
Problem Number: 517 URL: https://leetcode.com/problems/check-if-the-number-is-fascinating Title: 2729. Check if The Number is Fascinating Problem Description: You are given an integer n that consists of exactly 3 digits. We call the number n fascinating if, after the following modification, the resulting number contains all the digits from 1 to 9 exactly once and does not contain any 0's:

Concatenate n with the numbers 2 * n and 3 * n.

Return true if n is fascinating, or false otherwise. Concatenating two numbers means joining them together. For example, the concatenation of 121 and 371 is 121371. Example 1: Input: n = 192 Output: true Explanation: We concatenate the numbers n = 192 and 2 * n = 384 and 3 * n = 576. The resulting number is 192384576. This number contains all the digits from 1 to 9 exactly once.

Example 2: Input: n = 100 Output: false Explanation: We concatenate the numbers n = 100 and 2 * n = 200 and 3 * n = 300. The resulting number is 100200300. This number does not satisfy any of the conditions.

Constraints:

100 <= n <= 999

==
Problem Number: 518 URL: https://leetcode.com/problems/neither-minimum-nor-maximum Title: 2733. Neither Minimum nor Maximum Problem Description: Given an integer array nums containing distinct positive integers, find and return any number from the array that is neither the minimum nor the maximum value in the array, or -1 if there is no such number. Return the selected integer. Example 1: Input: nums = [3,2,1,4] Output: 2 Explanation: In this example, the minimum value is 1 and the maximum value is 4. Therefore, either 2 or 3 can be valid answers.

Example 2: Input: nums = [1,2] Output: -1 Explanation: Since there is no number in nums that is neither the maximum nor the minimum, we cannot select a number that satisfies the given condition. Therefore, there is no answer.

Example 3: Input: nums = [2,1,3] Output: 2 Explanation: Since 2 is neither the maximum nor the minimum value in nums, it is the only valid answer.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100 All values in nums are distinct

===

Problem Number: 519 URL: https://leetcode.com/problems/total-distance-traveled Title: 2739. Total Distance Traveled Problem Description: A truck has two fuel tanks. You are given two integers, mainTank representing the fuel present in the main tank in liters and additionalTank representing the fuel present in the additional tank in liters. The truck has a mileage of 10 km per liter. Whenever 5 liters of fuel get used up in the main tank, if the additional tank has at least 1 liters of fuel, 1 liters of fuel will be transferred from the additional tank to the main tank. Return the maximum distance which can be traveled. Note: Injection from the additional tank is not continuous. It happens suddenly and immediately for every 5 liters consumed. Example 1: Input: mainTank = 5, additionalTank = 10 Output: 60 Explanation: After spending 5 litre of fuel, fuel remaining is (5 - 5 + 1) = 1 litre and distance traveled is 50km. After spending another 1 litre of fuel, no fuel gets injected in the main tank and the main tank becomes empty. Total distance traveled is 60km.

Example 2: Input: mainTank = 1, additionalTank = 2 Output: 10 Explanation: After spending 1 litre of fuel, the main tank becomes empty. Total distance traveled is 10km.

Constraints:

1 <= mainTank, additionalTank <= 100

===

Problem Number: 520 URL: https://leetcode.com/problems/find-maximum-number-of-string-pairs Title: 2744. Find Maximum Number of String Pairs Problem Description: You are given a 0-indexed array words consisting of distinct strings. The string words[i] can be paired with the string words[j] if:

The string words[i] is equal to the reversed string of words[j]. 0 <= i < j < words.length.

Return the maximum number of pairs that can be formed from the array words. Note that each string can belong in at most one pair. Example 1: Input: words = ["cd","ac","dc","ca","zz"] Output: 2 Explanation: In this example, we can form 2 pair of strings in the following way: - We pair the 0th string with the 2nd string, as the reversed string of word[0] is "dc" and is equal to words[2]. - We pair the 1st string with the 3rd string, as the reversed string of word[1] is "ca" and is equal to words[3]. It can be proven that 2 is the maximum number of pairs that can be formed. Example 2: Input: words = ["ab","ba","cc"] Output: 1 Explanation: In this example, we can form 1 pair of strings in the following way: - We pair the 0th string with the 1st string, as the reversed string of words[1] is "ab" and is equal to words[0]. It can be proven that 1 is the maximum number of pairs that can be formed.

Example 3: Input: words = ["aa","ab"] Output: 0 Explanation: In this example, we are unable to form any pair of strings.

Constraints:

1 <= words.length <= 50 words[i].length == 2 words consists of distinct strings. words[i] contains only lowercase English letters.

==
Problem Number: 521 URL: https://leetcode.com/problems/number-of-beautiful-pairs Title: 2748. Number of Beautiful Pairs Problem Description: You are given a 0-indexed integer array nums. A pair of indices i, j where 0 <= i < j < nums.length is called beautiful if the first digit of nums[i] and the last digit of nums[j] are coprime. Return the total number of beautiful pairs in nums. Two integers x and y are coprime if there is no integer greater than 1 that divides both of them. In other words, x and y are coprime if gcd(x, y) == 1, where gcd(x, y) is the greatest common divisor of x and y. Example 1: Input: nums = [2,5,1,4] Output: 5 Explanation: There are 5 beautiful pairs in nums: When i = 0 and j = 1: the first digit of nums[0] is 2, and the last digit of nums[1] is 5. We can confirm that 2 and 5 are coprime, since gcd(2,5) == 1. When i = 0 and j = 2: the first digit of nums[0] is 2, and the last digit of nums[2] is 1. Indeed, gcd(2,1) == 1. When i = 1 and j = 2: the first digit of nums[1] is 5, and the last digit of nums[2] is 1. Indeed, gcd(5,1) == 1. When i = 1 and j = 3: the first digit of nums[1] is 5, and the last digit of nums[3] is 4. Indeed, gcd(5,4) == 1. When i = 2 and j = 3: the first digit of nums[2] is 1, and the last digit of nums[3] is 4. Indeed, gcd(1,4) == 1. Thus, we return 5.

Example 2: Input: nums = [11,21,12] Output: 2 Explanation: There are 2 beautiful pairs: When i = 0 and j = 1: the first digit of nums[0] is 1, and the last digit of nums[1] is 1. Indeed, gcd(1,1) == 1. When i = 0 and j = 2: the first digit of nums[0] is 1, and the last digit of nums[2] is 2. Indeed, gcd(1,2) == 1. Thus, we return 2.

Constraints:

2 <= nums.length <= 100 1 <= nums[i] <= 9999 nums[i] % 10 != 0

==
Problem Number: 522 URL: https://leetcode.com/problems/longest-even-odd-subarray-with-threshold Title: 2760. Longest Even Odd Subarray With Threshold Problem Description: You are given a 0-indexed integer array nums and an integer threshold. Find the length of the longest subarray of nums starting at index l and ending at index r (0 <= l <= r < nums.length) that satisfies the following conditions:

nums[l] % 2 == 0 For all indices i in the range [l, r - 1], nums[i] % 2 != nums[i + 1] % 2 For all indices i in the range [l, r], nums[i] <= threshold

Return an integer denoting the length of the longest such subarray. Note: A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [3,2,5,4], threshold = 5 Output: 3 Explanation: In this example, we can select the subarray that starts at l = 1 and ends at r = 3 => [2,5,4]. This subarray satisfies the conditions. Hence, the answer is the length of the subarray, 3. We can show that 3 is the maximum possible achievable length. Example 2: Input: nums = [1,2], threshold = 2 Output: 1 Explanation: In this example, we can select the subarray that starts at l = 1 and ends at r = 1 => [2]. It satisfies all the conditions and we can show that 1 is the maximum possible achievable length.

Example 3: Input: nums = [2,3,4,5], threshold = 4 Output: 3 Explanation: In this example, we can select the subarray that starts at l = 0 and ends at r = 2 => [2,3,4]. It satisfies all the conditions. Hence, the answer is the length of the subarray, 3. We can show that 3 is the maximum possible achievable length.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100 1 <= threshold <= 100

==
Problem Number: 523 URL: https://leetcode.com/problems/longest-alternating-subarray Title: 2765. Longest Alternating Subarray Problem Description: You are given a 0-indexed integer array nums. A subarray s of length m is called alternating if:

m is greater than 1. s1 = s0 + 1. The 0-indexed subarray s looks like [s0, s1, s0, s1,...,s(m-1) % 2]. In other words, s1 - s0 = 1, s2 - s1 = -1, s3 - s2 = 1, s4 - s3 = -1, and so on up to s[m - 1] - s[m - 2] = (-1)m.

Return the maximum length of all alternating subarrays present in nums or -1 if no such subarray exists. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [2,3,4,3,4] Output: 4 Explanation: The alternating subarrays are [3,4], [3,4,3], and [3,4,3,4]. The longest of these is [3,4,3,4], which is of length 4.

Example 2: Input: nums = [4,5,6] Output: 2 Explanation: [4,5] and [5,6] are the only two alternating subarrays. They are both of length 2.

Constraints:

2 <= nums.length <= 100 1 <= nums[i] <= 104

==
Problem Number: 524 URL: https://leetcode.com/problems/find-the-maximum-achievable-number Title: 2769. Find the Maximum Achievable Number Problem Description: You are given two integers, num and t. An integer x is called achievable if it can become equal to num after applying the following operation no more than t times:

Increase or decrease x by 1, and simultaneously increase or decrease num by 1.

Return the maximum possible achievable number. It can be proven that there exists at least one achievable number. Example 1: Input: num = 4, t = 1 Output: 6 Explanation: The maximum achievable number is x = 6; it can become equal to num after performing this operation: 1- Decrease x by 1, and increase num by 1. Now, x = 5 and num = 5. It can be proven that there is no achievable number larger than 6.

Example 2: Input: num = 3, t = 2 Output: 7 Explanation: The maximum achievable number is x = 7; after performing these operations, x will equal num: 1- Decrease x by 1, and increase num by 1. Now, x = 6 and num = 4. 2- Decrease x by 1, and increase num by 1. Now, x = 5 and num = 5. It can be proven that there is no achievable number larger than 7.

Constraints:

1 <= num, t <= 50

==
Problem Number: 525 URL: https://leetcode.com/problems/sum-of-squares-of-special-elements Title: 2778. Sum of Squares of Special Elements Problem Description: You are given a 1-indexed integer array nums of length n. An element nums[i] of nums is called special if i divides n, i.e. n % i == 0. Return the sum of the squares of all special elements of nums. Example 1: Input: nums = [1,2,3,4] Output: 21 Explanation: There are exactly 3 special elements in nums: nums[1] since 1 divides 4, nums[2] since 2 divides 4, and nums[4] since 4 divides 4. Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[4] * nums[4] = 1 * 1 + 2 * 2 + 4 * 4 = 21.

Example 2: Input: nums = [2,7,1,19,18,3] Output: 63 Explanation: There are exactly 4 special elements in nums: nums[1] since 1 divides 6, nums[2] since 2 divides 6, nums[3] since 3 divides 6, and nums[6] since 6 divides 6. Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[3] * nums[3] + nums[6] * nums[6] = 2 * 2 + 7 * 7 + 1 * 1 + 3 * 3 = 63.

Constraints:

1 <= nums.length == n <= 50 1 <= nums[i] <= 50

==
Problem Number: 526 URL: https://leetcode.com/problems/check-if-array-is-good Title: 2784. Check if Array is Good Problem Description: You are given an integer array nums. We consider an array good if it is a permutation of an array base[n]. base[n] = [1, 2, ..., n - 1, n, n] (in other words, it is an array of length n + 1 which contains 1 to n - 1 exactly once, plus two occurrences of n). For example, base[1] = [1, 1] and base[3] = [1, 2, 3, 3]. Return true if the given array is good, otherwise return false. Note: A permutation of integers represents an arrangement of these numbers. Example 1: Input: nums = [2, 1, 3] Output: false Explanation: Since the maximum element of the array is 3, the only candidate n for which this array could be a permutation of base[n], is n = 3. However, base[3] has four elements but array nums has three. Therefore, it can not be a permutation of base[3] = [1, 2, 3, 3]. So the answer is false.

Example 2: Input: nums = [1, 3, 3, 2] Output: true Explanation: Since the maximum element of the array is 3, the only candidate n for which this array could be a permutation of base[n], is n = 3. It can be seen that nums is a permutation of base[3] = [1, 2, 3, 3] (by swapping the second and fourth elements in nums, we reach base[3]). Therefore, the answer is true. Example 3: Input: nums = [1, 1] Output: true Explanation: Since the maximum element of the array is 1, the only candidate n for which this array could be a permutation of base[n], is n = 1. It can be seen that nums is a permutation of base[1] = [1, 1]. Therefore, the answer is true. Example 4: Input: nums = [3, 4, 4, 1, 2, 1] Output: false Explanation: Since the maximum element of the array is 4, the only candidate n for which this array could be a permutation of base[n], is n = 4. However, base[4] has five elements but array nums has six. Therefore, it can not be a permutation of base[4] = [1, 2, 3, 4, 4]. So the answer is false.

Constraints:

1 <= nums.length <= 100 1 <= num[i] <= 200

==

Problem Number: 527 URL: https://leetcode.com/problems/split-strings-by-separator Title: 2788. Split Strings by Separator Problem Description: Given an array of strings words and a character separator, split each string in words by separator. Return an array of strings containing the new strings formed after the splits, excluding empty strings. Notes

separator is used to determine where the split should occur, but it is not included as part of the resulting strings. A split may result in more than two strings. The resulting strings must maintain the same order as they were initially given.

Example 1: Input: words = ["one.two.three","four.five","six"], separator = "." Output: ["one","two","three","four","five","six"] Explanation: In this example we split as follows:

"one.two.three" splits into "one", "two", "three" "four.five" splits into "four", "five" "six" splits into "six"

Hence, the resulting array is ["one","two","three","four","five","six"]. Example 2: Input: words = ["$easy$","$problem$"], separator = "$" Output: ["easy","problem"] Explanation: In this example we split as follows:

"$easy$" splits into "easy" (excluding empty strings) "$problem$" splits into "problem" (excluding empty strings)

Hence, the resulting array is ["easy","problem"].

Example 3: Input: words = ["|||"], separator = "|" Output: [] Explanation: In this example the resulting split of "|||" will contain only empty strings, so we return an empty array []. Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 20 characters in words[i] are either lowercase English letters or characters from the string ".,|$#@" (excluding

the quotes) separator is a character from the string ".,|$#@" (excluding the quotes)

==

Problem Number: 528 URL: https://leetcode.com/problems/number-of-employees-who-met-the-target Title: 2798. Number of Employees Who Met the Target Problem Description: There are n employees in a company, numbered from 0 to n - 1. Each employee i has worked for hours[i] hours in the company. The company requires each employee to work for at least target hours. You are given a 0-indexed array of non-negative integers hours of length n and a non-negative integer target. Return the integer denoting the number of employees who worked at least target hours. Example 1: Input: hours = [0,1,2,3,4], target = 2 Output: 3 Explanation: The company wants each employee to work for at least 2 hours. - Employee 0 worked for 0 hours and didn't meet the target. - Employee 1 worked for 1 hours and didn't meet the target. - Employee 2 worked for 2 hours and met the target. - Employee 3 worked for 3 hours and met the target. - Employee 4 worked for 4 hours and met the target. There are 3 employees who met the target.

Example 2: Input: hours = [5,1,4,2,2], target = 6 Output: 0 Explanation: The company wants each employee to work for at least 6 hours. There are 0 employees who met the target.

Constraints:

1 <= n == hours.length <= 50 0 <= hours[i], target <= 105

==

Problem Number: 529 URL: https://leetcode.com/problems/account-balance-after-rounded-purchase Title: 2806. Account Balance After Rounded Purchase Problem Description: Initially, you have a bank account balance of 100 dollars. You are given an integer purchaseAmount representing the amount you will spend on a purchase in dollars. At the store where you will make the purchase, the purchase amount is rounded to the nearest multiple of 10. In other words, you pay a non-negative amount, roundedAmount, such that roundedAmount is a multiple of 10 and abs(roundedAmount - purchaseAmount) is minimized. If there is more than one nearest multiple of 10, the largest multiple is chosen. Return an integer denoting your account balance after making a purchase worth purchaseAmount dollars from the store. Note: 0 is considered to be a multiple of 10 in this problem. Example 1: Input: purchaseAmount = 9 Output: 90 Explanation: In this example, the nearest multiple of 10 to 9 is 10. Hence, your account balance becomes 100 - 10 = 90.

Example 2: Input: purchaseAmount = 15 Output: 80 Explanation: In this example, there are two nearest multiples of 10 to 15: 10 and 20. So, the larger multiple, 20, is chosen. Hence, your account balance becomes 100 - 20 = 80.

Constraints:

0 <= purchaseAmount <= 100

==

Problem Number: 530 URL: https://leetcode.com/problems/faulty-keyboard Title: 2810. Faulty Keyboard Problem Description: Your laptop keyboard is faulty, and whenever you type a character ’i’ on it, it reverses the string that you have written. Typing other characters works as expected. You are given a 0-indexed string s, and you type each character of s using your faulty keyboard. Return the final string that will be present on your laptop screen. Example 1: Input: s = ”string” Output: ”rtsng” Explanation: After typing first character, the text on the screen is ”s”. After the second character, the text is ”st”. After the third character, the text is ”str”. Since the fourth character is an ’i’, the text gets reversed and becomes ”rts”. After the fifth character, the text is ”rtsn”. After the sixth character, the text is ”rtsng”. Therefore, we return ”rtsng”.

Example 2: Input: s = ”poiinter” Output: ”ponter” Explanation: After the first character, the text on the screen is ”p”. After the second character, the text is ”po”. Since the third character you type is an ’i’, the text gets reversed and becomes ”op”. Since the fourth character you type is an ’i’, the text gets reversed and becomes ”po”. After the fifth character, the text is ”pon”. After the sixth character, the text is ”pont”. After the seventh character, the text is ”ponte”. After the eighth character, the text is ”ponter”. Therefore, we return ”ponter”. Constraints:

1 <= s.length <= 100 s consists of lowercase English letters. s[0] != ’i’

==

Problem Number: 531 URL: https://leetcode.com/problems/max-pair-sum-in-an-array Title: 2815. Max Pair Sum in an Array Problem Description: You are given a 0-indexed integer array nums. You have to find the maximum sum of a pair of numbers from nums such that the maximum digit in both numbers are equal. Return the maximum sum or -1 if no such pair exists. Example 1: Input: nums = [51,71,17,24,42] Output: 88 Explanation: For i = 1 and j = 2, nums[i] and nums[j] have equal maximum digits with a pair sum of 71 + 17 = 88. For i = 3 and j = 4, nums[i] and nums[j] have equal maximum digits with a pair sum of 24 + 42 = 66. It can be shown that there are no other pairs with equal maximum digits, so the answer is 88. Example 2: Input: nums = [1,2,3,4] Output: -1 Explanation: No pair exists in nums with equal maximum digits.

Constraints:

2 <= nums.length <= 100 1 <= nums[i] <= 104

==

Problem Number: 532 URL: https://leetcode.com/problems/count-pairs-whose-sum-is-less-than-target Title: 2824. Count Pairs Whose Sum is Less than Target Problem Description: Given a 0-indexed integer array nums of length n and an integer target, return the number of pairs (i, j) where 0 <= i < j < n and nums[i] + nums[j] < target. Example 1: Input: nums = [-1,1,2,3,1], target = 2 Output: 3 Explanation: There are 3 pairs of indices that satisfy the conditions in the statement: - (0, 1) since 0 < 1 and nums[0] + nums[1] = 0 < target - (0, 2) since 0 < 2 and nums[0] + nums[2] = 1 < target - (0, 4) since 0 < 4 and nums[0] + nums[4] = 0 < target Note that (0, 3) is not counted since nums[0] + nums[3] is not strictly less than the target.

Example 2: Input: nums = [-6,2,5,-2,-7,-1,3], target = -2 Output: 10 Explanation: There are 10 pairs of indices that satisfy the conditions in the statement: - (0, 1) since 0 < 1 and nums[0] + nums[1] = -4 < target - (0, 3) since 0 < 3 and nums[0] + nums[3] = -8 < target - (0, 4) since 0 < 4 and nums[0] + nums[4] = -13 < target - (0, 5) since 0 < 5 and nums[0] + nums[5] = -7 < target - (0, 6) since 0 < 6 and nums[0] + nums[6] = -3 < target - (1, 4) since 1 < 4 and nums[1] + nums[4] = -5 < target - (3, 4) since 3 < 4 and nums[3] + nums[4] = -9 < target - (3, 5) since 3 < 5 and nums[3] + nums[5] = -3 < target - (4, 5) since 4 < 5 and nums[4] + nums[5] = -8 < target - (4, 6) since 4 < 6 and nums[4] + nums[6] = -4 < target

Constraints:

1 <= nums.length == n <= 50 -50 <= nums[i], target <= 50

==
Problem Number: 533 URL: https://leetcode.com/problems/check-if-a-string-is-an-acronym-of-words Title: 2828. Check if a String Is an Acronym of Words Problem Description: Given an array of strings words and a string s, determine if s is an acronym of words. The string s is considered an acronym of words if it can be formed by concatenating the first character of each string in words in order. For example, "ab" can be formed from ["apple", "banana"], but it can't be formed from ["bear", "aardvark"]. Return true if s is an acronym of words, and false otherwise. Example 1: Input: words = ["alice","bob","charlie"], s = "abc" Output: true Explanation: The first character in the words "alice", "bob", and "charlie" are 'a', 'b', and 'c', respectively. Hence, s = "abc" is the acronym.

Example 2: Input: words = ["an","apple"], s = "a" Output: false Explanation: The first character in the words "an" and "apple" are 'a' and 'a', respectively. The acronym formed by concatenating these characters is "aa". Hence, s = "a" is not the acronym.

Example 3: Input: words = ["never","gonna","give","up","on","you"], s = "ngguoy" Output: true Explanation: By concatenating the first character of the words in the array, we get the string "ngguoy". Hence, s = "ngguoy" is the acronym.

Constraints:

1 <= words.length <= 100 1 <= words[i].length <= 10 1 <= s.length <= 100 words[i] and s consist of lowercase English letters.

==
Problem Number: 534 URL: https://leetcode.com/problems/furthest-point-

from-origin Title: 2833. Furthest Point From Origin Problem Description: You are given a string moves of length n consisting only of characters 'L', 'R', and '_'. The string represents your movement on a number line starting from the origin 0. In the ith move, you can choose one of the following directions:

move to the left if moves[i] = 'L' or moves[i] = '_' move to the right if moves[i] = 'R' or moves[i] = '_'

Return the distance from the origin of the furthest point you can get to after n moves. Example 1: Input: moves = "L_RL__R" Output: 3 Explanation: The furthest point we can reach from the origin 0 is point -3 through the following sequence of moves "LLRLLLR".

Example 2: Input: moves = "_R__LL_" Output: 5 Explanation: The furthest point we can reach from the origin 0 is point -5 through the following sequence of moves "LRLLLLL".

Example 3: Input: moves = "_______" Output: 7 Explanation: The furthest point we can reach from the origin 0 is point 7 through the following sequence of moves "RRRRRRR".

Constraints:

1 <= moves.length == n <= 50 moves consists only of characters 'L', 'R' and '_'.

==
Problem Number: 535 URL: https://leetcode.com/problems/check-if-strings-can-be-made-equal-with-operations-i Title: 2839. Check if Strings Can be Made Equal With Operations I Problem Description: You are given two strings s1 and s2, both of length 4, consisting of lowercase English letters. You can apply the following operation on any of the two strings any number of times:

Choose any two indices i and j such that j - i = 2, then swap the two characters at those indices in the string.

Return true if you can make the strings s1 and s2 equal, and false otherwise. Example 1: Input: s1 = "abcd", s2 = "cdab" Output: true Explanation: We can do the following operations on s1: - Choose the indices i = 0, j = 2. The resulting string is s1 = "cbad". - Choose the indices i = 1, j = 3. The resulting string is s1 = "cdab" = s2.

Example 2: Input: s1 = "abcd", s2 = "dacb" Output: false Explanation: It is not possible to make the two strings equal.

Constraints:

s1.length == s2.length == 4 s1 and s2 consist only of lowercase English letters.

==
Problem Number: 536 URL: https://leetcode.com/problems/count-symmetric-integers Title: 2843. Count Symmetric Integers Problem Description: You are

given two positive integers low and high. An integer x consisting of 2 * n digits is symmetric if the sum of the first n digits of x is equal to the sum of the last n digits of x. Numbers with an odd number of digits are never symmetric. Return the number of symmetric integers in the range [low, high].   Example 1: Input: low = 1, high = 100 Output: 9 Explanation: There are 9 symmetric integers between 1 and 100: 11, 22, 33, 44, 55, 66, 77, 88, and 99.

Example 2: Input: low = 1200, high = 1230 Output: 4 Explanation: There are 4 symmetric integers between 1200 and 1230: 1203, 1212, 1221, and 1230.

Constraints:

1 <= low <= high <= 104

==
Problem Number: 537 URL: https://leetcode.com/problems/points-that-intersect-with-cars Title: 2848. Points That Intersect With Cars Problem Description: You are given a 0-indexed 2D integer array nums representing the coordinates of the cars parking on a number line. For any index i, nums[i] = [starti, endi] where starti is the starting point of the ith car and endi is the ending point of the ith car. Return the number of integer points on the line that are covered with any part of a car.   Example 1: Input: nums = [[3,6],[1,5],[4,7]] Output: 7 Explanation: All the points from 1 to 7 intersect at least one car, therefore the answer would be 7.

Example 2: Input: nums = [[1,3],[5,8]] Output: 7 Explanation: Points intersecting at least one car are 1, 2, 3, 5, 6, 7, 8. There are a total of 7 points, therefore the answer would be 7.

Constraints:

1 <= nums.length <= 100 nums[i].length == 2 1 <= starti <= endi <= 100

==
Problem Number: 538 URL: https://leetcode.com/problems/minimum-right-shifts-to-sort-the-array Title: 2855. Minimum Right Shifts to Sort the Array Problem Description: You are given a 0-indexed array nums of length n containing distinct positive integers. Return the minimum number of right shifts required to sort nums and -1 if this is not possible. A right shift is defined as shifting the element at index i to index (i + 1) % n, for all indices. Example 1: Input: nums = [3,4,5,1,2] Output: 2 Explanation: After the first right shift, nums = [2,3,4,5,1]. After the second right shift, nums = [1,2,3,4,5]. Now nums is sorted; therefore the answer is 2.

Example 2: Input: nums = [1,3,5] Output: 0 Explanation: nums is already sorted therefore, the answer is 0. Example 3: Input: nums = [2,1,4] Output: -1 Explanation: It's impossible to sort the array using right shifts.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 100 nums contains distinct integers.

==
Problem Number: 539 URL: https://leetcode.com/problems/sum-of-values-at-indices-with-k-set-bits Title: 2859. Sum of Values at Indices With K Set Bits Problem Description: You are given a 0-indexed integer array nums and an integer k. Return an integer that denotes the sum of elements in nums whose corresponding indices have exactly k set bits in their binary representation. The set bits in an integer are the 1's present when it is written in binary.

For example, the binary representation of 21 is 10101, which has 3 set bits.

Example 1: Input: nums = [5,10,1,5,2], k = 1 Output: 13 Explanation: The binary representation of the indices are: 0 = 0002 1 = 0012 2 = 0102 3 = 0112 4 = 1002 Indices 1, 2, and 4 have k = 1 set bits in their binary representation. Hence, the answer is nums[1] + nums[2] + nums[4] = 13. Example 2: Input: nums = [4,3,2,1], k = 2 Output: 1 Explanation: The binary representation of the indices are: 0 = 002 1 = 012 2 = 102 3 = 112 Only index 3 has k = 2 set bits in its binary representation. Hence, the answer is nums[3] = 1.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 105 0 <= k <= 10

==
Problem Number: 540 URL: https://leetcode.com/problems/maximum-odd-binary-number Title: 2864. Maximum Odd Binary Number Problem Description: You are given a binary string s that contains at least one '1'. You have to rearrange the bits in such a way that the resulting binary number is the maximum odd binary number that can be created from this combination. Return a string representing the maximum odd binary number that can be created from the given combination. Note that the resulting string can have leading zeros. Example 1: Input: s = "010" Output: "001" Explanation: Because there is just one '1', it must be in the last position. So the answer is "001".

Example 2: Input: s = "0101" Output: "1001" Explanation: One of the '1's must be in the last position. The maximum number that can be made with the remaining digits is "100". So the answer is "1001".

Constraints:

1 <= s.length <= 100 s consists only of '0' and '1'. s contains at least one '1'.

==
Problem Number: 541 URL: https://leetcode.com/problems/add-two-numbers Title: 2. Add Two Numbers Problem Description: You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list. You may assume the two numbers do not contain any leading zero, except the number 0 itself. Example 1:

Input: l1 = [2,4,3], l2 = [5,6,4] Output: [7,0,8] Explanation: 342 + 465 = 807.

Example 2: Input: l1 = [0], l2 = [0] Output: [0]

Example 3: Input: l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9] Output: [8,9,9,9,0,0,0,1]

Constraints:

The number of nodes in each linked list is in the range [1, 100]. 0 <= Node.val <= 9 It is guaranteed that the list represents a number that does not have leading zeros.

==

Problem Number: 542 URL: https://leetcode.com/problems/longest-substring-without-repeating-characters Title: 3. Longest Substring Without Repeating Characters Problem Description: Given a string s, find the length of the longest substring without repeating characters. Example 1: Input: s = ”abcabcbb” Output: 3 Explanation: The answer is ”abc”, with the length of 3.

Example 2: Input: s = ”bbbbb” Output: 1 Explanation: The answer is ”b”, with the length of 1.

Example 3: Input: s = ”pwwkew” Output: 3 Explanation: The answer is ”wke”, with the length of 3. Notice that the answer must be a substring, ”pwke” is a subsequence and not a substring.

Constraints:

0 <= s.length <= 5 * 104 s consists of English letters, digits, symbols and spaces.

==

Problem Number: 543 URL: https://leetcode.com/problems/longest-palindromic-substring Title: 5. Longest Palindromic Substring Problem Description: Given a string s, return the longest palindromic substring in s. Example 1: Input: s = ”babad” Output: ”bab” Explanation: ”aba” is also a valid answer.

Example 2: Input: s = ”cbbd” Output: ”bb”

Constraints:

1 <= s.length <= 1000 s consist of only digits and English letters.

==

Problem Number: 544 URL: https://leetcode.com/problems/zigzag-conversion Title: 6. Zigzag Conversion Problem Description: The string ”PAYPALISHIRING” is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility) P A H N A P L S I I G Y I R

And then read line by line: ”PAHNAPLSIIGYIR” Write the code that will take a string and make this conversion given a number of rows: string convert(string s, int numRows);

Example 1: Input: s = ”PAYPALISHIRING”, numRows = 3 Output: ”PAHNAPLSIIGYIR”

Example 2: Input: s = ”PAYPALISHIRING”, numRows = 4 Output: ”PINALSIGYAHRPI” Explanation: P I N A L S I G Y A H R P I

Example 3: Input: s = ”A”, numRows = 1 Output: ”A”

Constraints:

1 <= s.length <= 1000 s consists of English letters (lower-case and upper-case), ’,’ and ’.’. 1 <= numRows <= 1000

==
Problem Number: 545 URL: https://leetcode.com/problems/reverse-integer Title: 7. Reverse Integer Problem Description: Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-231, 231 - 1], then return 0. Assume the environment does not allow you to store 64-bit integers (signed or unsigned). Example 1: Input: x = 123 Output: 321

Example 2: Input: x = -123 Output: -321

Example 3: Input: x = 120 Output: 21

Constraints:

-231 <= x <= 231 - 1

==
Problem Number: 546 URL: https://leetcode.com/problems/string-to-integer-atoi Title: 8. String to Integer (atoi) Problem Description: Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer (similar to C/C++’s atoi function). The algorithm for myAtoi(string s) is as follows:

Read in and ignore any leading whitespace. Check if the next character (if not already at the end of the string) is ’-’ or ’+’. Read this character in if it is either. This determines if the final result is negative or positive respectively. Assume the result is positive if neither is present. Read in next the characters until the next non-digit character or the end of the input is reached. The rest of the string is ignored. Convert these digits into an integer (i.e. ”123” -> 123, ”0032” -> 32). If no digits were read, then the integer is 0. Change the sign as necessary (from step 2). If the integer is out of the 32-bit signed integer range [-231, 231 - 1], then clamp the integer so that it remains in the range. Specifically, integers less than -231 should be clamped to -231, and integers greater than 231 - 1 should be clamped to 231 - 1. Return the integer as the final result.

Note:

Only the space character ’ ’ is considered a whitespace character. Do not ignore any characters other than the leading whitespace or the rest of the string after the digits.

Example 1: Input: s = "42" Output: 42 Explanation: The underlined characters are what is read in, the caret is the current reader position. Step 1: "42" (no characters read because there is no leading whitespace) ^ Step 2: "42" (no characters read because there is neither a '-' nor '+') ^ Step 3: "42" ("42" is read in) ^ The parsed integer is 42. Since 42 is in the range [-231, 231 - 1], the final result is 42.

Example 2: Input: s = " -42" Output: -42 Explanation: Step 1: " -42" (leading whitespace is read and ignored) ^ Step 2: " -42" ('-' is read, so the result should be negative) ^ Step 3: " -42" ("42" is read in) ^ The parsed integer is -42. Since -42 is in the range [-231, 231 - 1], the final result is -42.

Example 3: Input: s = "4193 with words" Output: 4193 Explanation: Step 1: "4193 with words" (no characters read because there is no leading whitespace) ^ Step 2: "4193 with words" (no characters read because there is neither a '-' nor '+') ^ Step 3: "4193 with words" ("4193" is read in; reading stops because the next character is a non-digit) ^ The parsed integer is 4193. Since 4193 is in the range [-231, 231 - 1], the final result is 4193.

Constraints:

0 <= s.length <= 200 s consists of English letters (lower-case and upper-case), digits (0-9), ' ', '+', '-', and '.'.

==
Problem Number: 547 URL: https://leetcode.com/problems/container-with-most-water Title: 11. Container With Most Water Problem Description: You are given an integer array height of length n. There are n vertical lines drawn such that the two endpoints of the ith line are (i, 0) and (i, height[i]). Find two lines that together with the x-axis form a container, such that the container contains the most water. Return the maximum amount of water a container can store. Notice that you may not slant the container. Example 1:

Input: height = [1,8,6,2,5,4,8,3,7] Output: 49 Explanation: The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

Example 2: Input: height = [1,1] Output: 1

Constraints:

n == height.length 2 <= n <= 105 0 <= height[i] <= 104

==
Problem Number: 548 URL: https://leetcode.com/problems/integer-to-roman Title: 12. Integer to Roman Problem Description: Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M. Symbol Value I 1 V 5 X 10 L 50 C 100 D 500 M 1000 For example, 2 is written as II in Roman numeral, just two one's added together. 12 is written as XII, which

is simply X + II. The number 27 is written as XXVII, which is XX + V + II. Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

I can be placed before V (5) and X (10) to make 4 and 9. X can be placed before L (50) and C (100) to make 40 and 90. C can be placed before D (500) and M (1000) to make 400 and 900.

Given an integer, convert it to a roman numeral. Example 1: Input: num = 3 Output: "III" Explanation: 3 is represented as 3 ones.

Example 2: Input: num = 58 Output: "LVIII" Explanation: L = 50, V = 5, III = 3.

Example 3: Input: num = 1994 Output: "MCMXCIV" Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

Constraints:

1 <= num <= 3999

==

Problem Number: 549 URL: https://leetcode.com/problems/3sum Title: 15. 3Sum Problem Description: Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0. Notice that the solution set must not contain duplicate triplets. Example 1: Input: nums = [-1,0,1,2,-1,-4] Output: [[-1,-1,2],[-1,0,1]] Explanation: nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0. nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0. nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0. The distinct triplets are [-1,0,1] and [-1,-1,2]. Notice that the order of the output and the order of the triplets does not matter.

Example 2: Input: nums = [0,1,1] Output: [] Explanation: The only possible triplet does not sum up to 0.

Example 3: Input: nums = [0,0,0] Output: [[0,0,0]] Explanation: The only possible triplet sums up to 0.

Constraints:

3 <= nums.length <= 3000 -105 <= nums[i] <= 105

==

Problem Number: 550 URL: https://leetcode.com/problems/3sum-closest Title: 16. 3Sum Closest Problem Description: Given an integer array nums of length n and an integer target, find three integers in nums such that the sum is closest to target. Return the sum of the three integers. You may assume that each input would have exactly one solution. Example 1: Input: nums =

[-1,2,1,-4], target = 1 Output: 2 Explanation: The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

Example 2: Input: nums = [0,0,0], target = 1 Output: 0 Explanation: The sum that is closest to the target is 0. (0 + 0 + 0 = 0).

Constraints:

3 <= nums.length <= 500 -1000 <= nums[i] <= 1000 -104 <= target <= 104

==
Problem Number: 551 URL: https://leetcode.com/problems/letter-combinations-of-a-phone-number Title: 17. Letter Combinations of a Phone Number Problem Description: Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order. A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

Example 1: Input: digits = "23" Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

Example 2: Input: digits = "" Output: []

Example 3: Input: digits = "2" Output: ["a","b","c"]

Constraints:

0 <= digits.length <= 4 digits[i] is a digit in the range ['2', '9'].

==
Problem Number: 552 URL: https://leetcode.com/problems/4sum Title: 18. 4Sum Problem Description: Given an array nums of n integers, return an array of all the unique quadruplets [nums[a], nums[b], nums[c], nums[d]] such that:

0 <= a, b, c, d < n a, b, c, and d are distinct. nums[a] + nums[b] + nums[c] + nums[d] == target

You may return the answer in any order. Example 1: Input: nums = [1,0,-1,0,-2,2], target = 0 Output: [[-2,-1,1,2],[-2,0,0,2],[-1,0,0,1]]

Example 2: Input: nums = [2,2,2,2,2], target = 8 Output: [[2,2,2,2]]

Constraints:

1 <= nums.length <= 200 -109 <= nums[i] <= 109 -109 <= target <= 109

==
Problem Number: 553 URL: https://leetcode.com/problems/remove-nth-node-from-end-of-list Title: 19. Remove Nth Node From End of List Problem Description: Given the head of a linked list, remove the nth node from the end of the list and return its head. Example 1:

Input: head = [1,2,3,4,5], n = 2 Output: [1,2,3,5]

Example 2: Input: head = [1], n = 1 Output: []

Example 3: Input: head = [1,2], n = 1 Output: [1]

Constraints:

The number of nodes in the list is sz. 1 <= sz <= 30 0 <= Node.val <= 100 1 <= n <= sz

Follow up: Could you do this in one pass?

==

Problem Number: 554 URL: https://leetcode.com/problems/generate-parentheses Title: 22. Generate Parentheses Problem Description: Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses. Example 1: Input: n = 3 Output: ["((()))","(()())","(())()","()(())","()()()"] Example 2: Input: n = 1 Output: ["()"]

Constraints:

1 <= n <= 8

==

Problem Number: 555 URL: https://leetcode.com/problems/swap-nodes-in-pairs Title: 24. Swap Nodes in Pairs Problem Description: Given a linked list, swap every two adjacent nodes and return its head. You must solve the problem without modifying the values in the list's nodes (i.e., only nodes themselves may be changed.) Example 1:

Input: head = [1,2,3,4] Output: [2,1,4,3]

Example 2: Input: head = [] Output: []

Example 3: Input: head = [1] Output: [1]

Constraints:

The number of nodes in the list is in the range [0, 100]. 0 <= Node.val <= 100

==

Problem Number: 556 URL: https://leetcode.com/problems/divide-two-integers Title: 29. Divide Two Integers Problem Description: Given two integers dividend and divisor, divide two integers without using multiplication, division, and mod operator. The integer division should truncate toward zero, which means losing its fractional part. For example, 8.345 would be truncated to 8, and -2.7335 would be truncated to -2. Return the quotient after dividing dividend by divisor. Note: Assume we are dealing with an environment that could only store integers within the 32-bit signed integer range: [−231, 231 − 1]. For this problem, if the quotient is strictly greater than 231 - 1, then return 231 - 1, and if the quotient is strictly less than -231, then return -231. Example 1: Input: dividend = 10, divisor = 3 Output: 3 Explanation: 10/3 = 3.33333.. which is truncated to 3.

Example 2: Input: dividend = 7, divisor = -3 Output: -2 Explanation: 7/-3 = -2.33333.. which is truncated to -2.

Constraints:

-231 <= dividend, divisor <= 231 - 1 divisor != 0

== Problem Number: 557 URL: https://leetcode.com/problems/next-permutation Title: 31. Next Permutation Problem Description: A permutation of an array of integers is an arrangement of its members into a sequence or linear order.

For example, for arr = [1,2,3], the following are all the permutations of arr: [1,2,3], [1,3,2], [2, 1, 3], [2, 3, 1], [3,1,2], [3,2,1].

The next permutation of an array of integers is the next lexicographically greater permutation of its integer. More formally, if all the permutations of the array are sorted in one container according to their lexicographical order, then the next permutation of that array is the permutation that follows it in the sorted container. If such arrangement is not possible, the array must be rearranged as the lowest possible order (i.e., sorted in ascending order).

For example, the next permutation of arr = [1,2,3] is [1,3,2]. Similarly, the next permutation of arr = [2,3,1] is [3,1,2]. While the next permutation of arr = [3,2,1] is [1,2,3] because [3,2,1] does not have a lexicographical larger rearrangement.

Given an array of integers nums, find the next permutation of nums. The replacement must be in place and use only constant extra memory. Example 1: Input: nums = [1,2,3] Output: [1,3,2]

Example 2: Input: nums = [3,2,1] Output: [1,2,3]

Example 3: Input: nums = [1,1,5] Output: [1,5,1]

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 100

== Problem Number: 558 URL: https://leetcode.com/problems/search-in-rotated-sorted-array Title: 33. Search in Rotated Sorted Array Problem Description: There is an integer array nums sorted in ascending order (with distinct values). Prior to being passed to your function, nums is possibly rotated at an unknown pivot index k (1 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,5,6,7] might be rotated at pivot index 3 and become [4,5,6,7,0,1,2]. Given the array nums after the possible rotation and an integer target, return the index of target if it is in nums, or -1 if it is not in nums. You must write an algorithm with O(log n) runtime complexity. Example 1: Input: nums = [4,5,6,7,0,1,2], target = 0 Output: 4 Example 2: Input: nums = [4,5,6,7,0,1,2], target = 3 Output: -1 Example 3: Input: nums = [1], target = 0 Output: -1

Constraints:

1 <= nums.length <= 5000 -104 <= nums[i] <= 104 All values of nums are unique. nums is an ascending array that is possibly rotated. -104 <= target <= 104

==
Problem Number: 559 URL: https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array Title: 34. Find First and Last Position of Element in Sorted Array Problem Description: Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value. If target is not found in the array, return [-1, -1]. You must write an algorithm with O(log n) runtime complexity. Example 1: Input: nums = [5,7,7,8,8,10], target = 8 Output: [3,4] Example 2: Input: nums = [5,7,7,8,8,10], target = 6 Output: [-1,-1] Example 3: Input: nums = [], target = 0 Output: [-1,-1]

Constraints:

0 <= nums.length <= 105 -109 <= nums[i] <= 109 nums is a non-decreasing array. -109 <= target <= 109

==
Problem Number: 560 URL: https://leetcode.com/problems/valid-sudoku Title: 36. Valid Sudoku Problem Description: Determine if a 9 x 9 Sudoku board is valid. Only the filled cells need to be validated according to the following rules:

Each row must contain the digits 1-9 without repetition. Each column must contain the digits 1-9 without repetition. Each of the nine 3 x 3 sub-boxes of the grid must contain the digits 1-9 without repetition.

Note:

A Sudoku board (partially filled) could be valid but is not necessarily solvable. Only the filled cells need to be validated according to the mentioned rules.

Example 1:

Input: board = [[”5”,”3”,”.”,”.”,”7”,”.”,”.”,”.”,”.”] ,[”6”,”.”,”.”,”1”,”9”,”5”,”.”,”.”,”.”] ,[”.”,”9”,”8”,”.”,”.”,”.”,”.”,”6”,”.”] ,[”8”,”.”,”.”,”.”,”6”,”.”,”.”,”.”,”3”] ,[”4”,”.”,”.”,”8”,”.”,”3”,”.”,”.”,”1”] ,[”7”,”.”,”.”,”.”,”2”,”.”,”.”,”.”,”6”] ,[”.”,”6”,”.”,”.”,”.”,”.”,”2”,”8”,”.”] ,[”.”,”.”,”.”,”4”,”1”,”9”,”.”,”.”,”5”] ,[”.”,”.”,”.”,”.”,”8”,”.”,”.”,”7”,”9”]] Output: true

Example 2: Input: board = [[”8”,”3”,”.”,”.”,”7”,”.”,”.”,”.”,”.”] ,[”6”,”.”,”.”,”1”,”9”,”5”,”.”,”.”,”.”] ,[”.”,”9”,”8”,”.”,”.”,”.”,”.”,”6”,”.”] ,[”8”,”.”,”.”,”.”,”6”,”.”,”.”,”.”,”3”] ,[”4”,”.”,”.”,”8”,”.”,”3”,”.”,”.”,”1”] ,[”7”,”.”,”.”,”.”,”2”,”.”,”.”,”.”,”6”] ,[”.”,”6”,”.”,”.”,”.”,”.”,”2”,”8”,”.”] ,[”.”,”.”,”.”,”4”,”1”,”9”,”.”,”.”,”5”] ,[”.”,”.”,”.”,”.”,”8”,”.”,”.”,”7”,”9”]] Output: false Explanation: Same as Example 1, except with the 5 in the top left corner being modified to 8. Since there are two 8’s in the top left 3x3 sub-box, it is invalid.

Constraints:

board.length == 9 board[i].length == 9 board[i][j] is a digit 1-9 or ”.”.

===

Problem Number: 561 URL: https://leetcode.com/problems/count-and-say Title: 38. Count and Say Problem Description: The count-and-say sequence is a sequence of digit strings defined by the recursive formula:

countAndSay(1) = ”1” countAndSay(n) is the way you would ”say” the digit string from countAndSay(n-1), which is then converted into a different digit string.

To determine how you ”say” a digit string, split it into the minimal number of substrings such that each substring contains exactly one unique digit. Then for each substring, say the number of digits, then say the digit. Finally, concatenate every said digit. For example, the saying and conversion for digit string ”3322251”:

Given a positive integer n, return the nth term of the count-and-say sequence. Example 1: Input: n = 1 Output: ”1” Explanation: This is the base case.

Example 2: Input: n = 4 Output: ”1211” Explanation: countAndSay(1) = ”1” countAndSay(2) = say ”1” = one 1 = ”11” countAndSay(3) = say ”11” = two 1’s = ”21” countAndSay(4) = say ”21” = one 2 + one 1 = ”12” + ”11” = ”1211”

Constraints:

1 <= n <= 30

===

Problem Number: 562 URL: https://leetcode.com/problems/combination-sum Title: 39. Combination Sum Problem Description: Given an array of distinct integers candidates and a target integer target, return a list of all unique combinations of candidates where the chosen numbers sum to target. You may return the combinations in any order. The same number may be chosen from candidates an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different. The test cases are generated such that the number of unique combinations that sum up to target is less than 150 combinations for the given input. Example 1: Input: candidates = [2,3,6,7], target = 7 Output: [[2,2,3],[7]] Explanation: 2 and 3 are candidates, and 2 + 2 + 3 = 7. Note that 2 can be used multiple times. 7 is a candidate, and 7 = 7. These are the only two combinations.

Example 2: Input: candidates = [2,3,5], target = 8 Output: [[2,2,2,2],[2,3,3],[3,5]]

Example 3: Input: candidates = [2], target = 1 Output: []

Constraints:

1 <= candidates.length <= 30 2 <= candidates[i] <= 40 All elements of candidates are distinct. 1 <= target <= 40

==
Problem Number: 563 URL: https://leetcode.com/problems/combination-sum-ii Title: 40. Combination Sum II Problem Description: Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target. Each number in candidates may only be used once in the combination. Note: The solution set must not contain duplicate combinations. Example 1: Input: candidates = [10,1,2,7,6,1,5], target = 8 Output: [[1,1,6], [1,2,5], [1,7], [2,6]]

Example 2: Input: candidates = [2,5,2,1,2], target = 5 Output: [[1,2,2], [5]]

Constraints:

1 <= candidates.length <= 100 1 <= candidates[i] <= 50 1 <= target <= 30
==
Problem Number: 564 URL: https://leetcode.com/problems/multiply-strings Title: 43. Multiply Strings Problem Description: Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2, also represented as a string. Note: You must not use any built-in BigInteger library or convert the inputs to integer directly. Example 1: Input: num1 = "2", num2 = "3" Output: "6" Example 2: Input: num1 = "123", num2 = "456" Output: "56088"

Constraints:

1 <= num1.length, num2.length <= 200 num1 and num2 consist of digits only. Both num1 and num2 do not contain any leading zero, except the number 0 itself.
==
Problem Number: 565 URL: https://leetcode.com/problems/jump-game-ii Title: 45. Jump Game II Problem Description: You are given a 0-indexed array of integers nums of length n. You are initially positioned at nums[0]. Each element nums[i] represents the maximum length of a forward jump from index i. In other words, if you are at nums[i], you can jump to any nums[i + j] where:

0 <= j <= nums[i] and i + j < n

Return the minimum number of jumps to reach nums[n - 1]. The test cases are generated such that you can reach nums[n - 1]. Example 1: Input: nums = [2,3,1,1,4] Output: 2 Explanation: The minimum number of jumps to reach the last index is 2. Jump 1 step from index 0 to 1, then 3 steps to the last index.

Example 2: Input: nums = [2,3,0,1,4] Output: 2

Constraints:

1 <= nums.length <= 104 0 <= nums[i] <= 1000 It's guaranteed that you can reach nums[n - 1].

==

Problem Number: 566 URL: https://leetcode.com/problems/permutations Title: 46. Permutations Problem Description: Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order. Example 1: Input: nums = [1,2,3] Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]] Example 2: Input: nums = [0,1] Output: [[0,1],[1,0]] Example 3: Input: nums = [1] Output: [[1]]

Constraints:

1 <= nums.length <= 6 -10 <= nums[i] <= 10 All the integers of nums are unique.

==

Problem Number: 567 URL: https://leetcode.com/problems/permutations-ii Title: 47. Permutations II Problem Description: Given a collection of numbers, nums, that might contain duplicates, return all possible unique permutations in any order. Example 1: Input: nums = [1,1,2] Output: [[1,1,2], [1,2,1], [2,1,1]]

Example 2: Input: nums = [1,2,3] Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

Constraints:

1 <= nums.length <= 8 -10 <= nums[i] <= 10

==

Problem Number: 568 URL: https://leetcode.com/problems/rotate-image Title: 48. Rotate Image Problem Description: You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise). You have to rotate the image in-place, which means you have to modify the input 2D matrix directly. DO NOT allocate another 2D matrix and do the rotation. Example 1:

Input: matrix = [[1,2,3],[4,5,6],[7,8,9]] Output: [[7,4,1],[8,5,2],[9,6,3]]

Example 2:

Input: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]] Output: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]

Constraints:

n == matrix.length == matrix[i].length 1 <= n <= 20 -1000 <= matrix[i][j] <= 1000

==

Problem Number: 569 URL: https://leetcode.com/problems/group-anagrams Title: 49. Group Anagrams Problem Description: Given an array of strings strs, group the anagrams together. You can return the answer in any order. An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once. Example 1: Input: strs = [”eat”,”tea”,”tan”,”ate”,”nat”,”bat”] Output:

[["bat"],["nat","tan"],["ate","eat","tea"]] Example 2: Input: strs = [""] Output: [[""]] Example 3: Input: strs = ["a"] Output: [["a"]]

Constraints:

1 <= strs.length <= 104 0 <= strs[i].length <= 100 strs[i] consists of lowercase English letters.

==

Problem Number: 570 URL: https://leetcode.com/problems/powx-n Title: 50. Pow(x, n) Problem Description: Implement pow(x, n), which calculates x raised to the power n (i.e., xn). Example 1: Input: x = 2.00000, n = 10 Output: 1024.00000

Example 2: Input: x = 2.10000, n = 3 Output: 9.26100

Example 3: Input: x = 2.00000, n = -2 Output: 0.25000 Explanation: 2-2 = 1/22 = 1/4 = 0.25

Constraints:

-100.0 < x < 100.0 -231 <= n <= 231-1 n is an integer. Either x is not zero or n > 0. -104 <= xn <= 104

==

Problem Number: 571 URL: https://leetcode.com/problems/maximum-subarray Title: 53. Maximum Subarray Problem Description: Given an integer array nums, find the subarray with the largest sum, and return its sum. Example 1: Input: nums = [-2,1,-3,4,-1,2,1,-5,4] Output: 6 Explanation: The subarray [4,-1,2,1] has the largest sum 6.

Example 2: Input: nums = [1] Output: 1 Explanation: The subarray [1] has the largest sum 1.

Example 3: Input: nums = [5,4,-1,7,8] Output: 23 Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104

Follow up: If you have figured out the O(n) solution, try coding another solution using the divide and conquer approach, which is more subtle.

==

Problem Number: 572 URL: https://leetcode.com/problems/spiral-matrix Title: 54. Spiral Matrix Problem Description: Given an m x n matrix, return all elements of the matrix in spiral order. Example 1:

Input: matrix = [[1,2,3],[4,5,6],[7,8,9]] Output: [1,2,3,6,9,8,7,4,5]

Example 2:

Input: matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]] Output: [1,2,3,4,8,12,11,10,9,5,6,7]

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 10 -100 <= matrix[i][j] <= 100

==
Problem Number: 573 URL: https://leetcode.com/problems/jump-game Title: 55. Jump Game Problem Description: You are given an integer array nums. You are initially positioned at the array's first index, and each element in the array represents your maximum jump length at that position. Return true if you can reach the last index, or false otherwise.   Example 1: Input: nums = [2,3,1,1,4] Output: true Explanation: Jump 1 step from index 0 to 1, then 3 steps to the last index.

Example 2: Input: nums = [3,2,1,0,4] Output: false Explanation: You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

Constraints:

1 <= nums.length <= 104 0 <= nums[i] <= 105

==
Problem Number: 574 URL: https://leetcode.com/problems/merge-intervals Title: 56. Merge Intervals Problem Description: Given an array of intervals where intervals[i] = [starti, endi], merge all overlapping intervals, and return an array of the non-overlapping intervals that cover all the intervals in the input.   Example 1: Input: intervals = [[1,3],[2,6],[8,10],[15,18]] Output: [[1,6],[8,10],[15,18]] Explanation: Since intervals [1,3] and [2,6] overlap, merge them into [1,6].

Example 2: Input: intervals = [[1,4],[4,5]] Output: [[1,5]] Explanation: Intervals [1,4] and [4,5] are considered overlapping.

Constraints:

1 <= intervals.length <= 104 intervals[i].length == 2 0 <= starti <= endi <= 104

==
Problem Number: 575 URL: https://leetcode.com/problems/insert-interval Title: 57. Insert Interval Problem Description: You are given an array of non-overlapping intervals intervals where intervals[i] = [starti, endi] represent the start and the end of the ith interval and intervals is sorted in ascending order by starti. You are also given an interval newInterval = [start, end] that represents the start and end of another interval. Insert newInterval into intervals such that intervals is still sorted in ascending order by starti and intervals still does not have any overlapping intervals (merge overlapping intervals if necessary). Return intervals after the insertion.   Example 1: Input: intervals = [[1,3],[6,9]], newInterval = [2,5] Output: [[1,5],[6,9]]

Example 2: Input: intervals = [[1,2],[3,5],[6,7],[8,10],[12,16]], newInterval = [4,8] Output: [[1,2],[3,10],[12,16]] Explanation: Because the new interval [4,8] overlaps with [3,5],[6,7],[8,10].

Constraints:

0 <= intervals.length <= 104 intervals[i].length == 2 0 <= starti <= endi <= 105 intervals is sorted by starti in ascending order. newInterval.length == 2 0 <= start <= end <= 105

==
Problem Number: 576 URL: https://leetcode.com/problems/spiral-matrix-ii Title: 59. Spiral Matrix II Problem Description: Given a positive integer n, generate an n x n matrix filled with elements from 1 to n2 in spiral order. Example 1:

Input: n = 3 Output: [[1,2,3],[8,9,4],[7,6,5]]

Example 2: Input: n = 1 Output: [[1]]

Constraints:

1 <= n <= 20

==
Problem Number: 577 URL: https://leetcode.com/problems/rotate-list Title: 61. Rotate List Problem Description: Given the head of a linked list, rotate the list to the right by k places. Example 1:

Input: head = [1,2,3,4,5], k = 2 Output: [4,5,1,2,3]

Example 2:

Input: head = [0,1,2], k = 4 Output: [2,0,1]

Constraints:

The number of nodes in the list is in the range [0, 500]. -100 <= Node.val <= 100 0 <= k <= 2 * 109

==
Problem Number: 578 URL: https://leetcode.com/problems/unique-paths Title: 62. Unique Paths Problem Description: There is a robot on an m x n grid. The robot is initially located at the top-left corner (i.e., grid[0][0]). The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time. Given the two integers m and n, return the number of possible unique paths that the robot can take to reach the bottom-right corner. The test cases are generated so that the answer will be less than or equal to 2 * 109. Example 1:

Input: m = 3, n = 7 Output: 28

Example 2: Input: m = 3, n = 2 Output: 3 Explanation: From the top-left corner, there are a total of 3 ways to reach the bottom-right corner: 1. Right -> Down -> Down 2. Down -> Down -> Right 3. Down -> Right -> Down

Constraints:

1 <= m, n <= 100

==
Problem Number: 579 URL: https://leetcode.com/problems/unique-paths-ii Title: 63. Unique Paths II Problem Description: You are given an m x n integer array grid. There is a robot initially located at the top-left corner (i.e., grid[0][0]). The robot tries to move to the bottom-right corner (i.e., grid[m - 1][n - 1]). The robot can only move either down or right at any point in time. An obstacle and space are marked as 1 or 0 respectively in grid. A path that the robot takes cannot include any square that is an obstacle. Return the number of possible unique paths that the robot can take to reach the bottom-right corner. The testcases are generated so that the answer will be less than or equal to 2 * 109. Example 1:

Input: obstacleGrid = [[0,0,0],[0,1,0],[0,0,0]] Output: 2 Explanation: There is one obstacle in the middle of the 3x3 grid above. There are two ways to reach the bottom-right corner: 1. Right -> Right -> Down -> Down 2. Down -> Down -> Right -> Right

Example 2:

Input: obstacleGrid = [[0,1],[0,0]] Output: 1

Constraints:

m == obstacleGrid.length n == obstacleGrid[i].length 1 <= m, n <= 100 obstacleGrid[i][j] is 0 or 1.

==
Problem Number: 580 URL: https://leetcode.com/problems/minimum-path-sum Title: 64. Minimum Path Sum Problem Description: Given a m x n grid filled with non-negative numbers, find a path from top left to bottom right, which minimizes the sum of all numbers along its path. Note: You can only move either down or right at any point in time. Example 1:

Input: grid = [[1,3,1],[1,5,1],[4,2,1]] Output: 7 Explanation: Because the path 1 → 3 → 1 → 1 → 1 minimizes the sum.

Example 2: Input: grid = [[1,2,3],[4,5,6]] Output: 12

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 200 0 <= grid[i][j] <= 200

==
Problem Number: 581 URL: https://leetcode.com/problems/simplify-path

Title: 71. Simplify Path Problem Description: Given a string path, which is an absolute path (starting with a slash '/') to a file or directory in a Unix-style file system, convert it to the simplified canonical path. In a Unix-style file system, a period '.' refers to the current directory, a double period '..' refers to the directory up a level, and any multiple consecutive slashes (i.e. '//') are treated as a single slash '/'. For this problem, any other format of periods such as '...' are treated as file/directory names. The canonical path should have the following format:

The path starts with a single slash '/'. Any two directories are separated by a single slash '/'. The path does not end with a trailing '/'. The path only contains the directories on the path from the root directory to the target file or directory (i.e., no period '.' or double period '..')

Return the simplified canonical path. Example 1: Input: path = "/home/" Output: "/home" Explanation: Note that there is no trailing slash after the last directory name.

Example 2: Input: path = "/../" Output: "/" Explanation: Going one level up from the root directory is a no-op, as the root level is the highest level you can go.

Example 3: Input: path = "/home//foo/" Output: "/home/foo" Explanation: In the canonical path, multiple consecutive slashes are replaced by a single one.

Constraints:

1 <= path.length <= 3000 path consists of English letters, digits, period '.', slash '/' or '_'. path is a valid absolute Unix path.

==

Problem Number: 582 URL: https://leetcode.com/problems/edit-distance Title: 72. Edit Distance Problem Description: Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2. You have the following three operations permitted on a word:

Insert a character Delete a character Replace a character

Example 1: Input: word1 = "horse", word2 = "ros" Output: 3 Explanation: horse -> rorse (replace 'h' with 'r') rorse -> rose (remove 'r') rose -> ros (remove 'e')

Example 2: Input: word1 = "intention", word2 = "execution" Output: 5 Explanation: intention -> inention (remove 't') inention -> enention (replace 'i' with 'e') enention -> exention (replace 'n' with 'x') exention -> exection (replace 'n' with 'c') exection -> execution (insert 'u')

Constraints:

0 <= word1.length, word2.length <= 500 word1 and word2 consist of lowercase English letters.

==
Problem Number: 583 URL: https://leetcode.com/problems/set-matrix-zeroes Title: 73. Set Matrix Zeroes Problem Description: Given an m x n integer matrix matrix, if an element is 0, set its entire row and column to 0's. You must do it in place. Example 1:

Input: matrix = [[1,1,1],[1,0,1],[1,1,1]] Output: [[1,0,1],[0,0,0],[1,0,1]]

Example 2:

Input: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]] Output: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]

Constraints:

m == matrix.length n == matrix[0].length 1 <= m, n <= 200 -231 <= matrix[i][j] <= 231 - 1

Follow up:

A straightforward solution using O(mn) space is probably a bad idea. A simple improvement uses O(m + n) space, but still not the best solution. Could you devise a constant space solution?

==
Problem Number: 584 URL: https://leetcode.com/problems/search-a-2d-matrix Title: 74. Search a 2D Matrix Problem Description: You are given an m x n integer matrix matrix with the following two properties:

Each row is sorted in non-decreasing order. The first integer of each row is greater than the last integer of the previous row.

Given an integer target, return true if target is in matrix or false otherwise. You must write a solution in O(log(m * n)) time complexity. Example 1:

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3 Output: true

Example 2:

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 13 Output: false

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 100 -104 <= matrix[i][j], target <= 104

==
Problem Number: 585 URL: https://leetcode.com/problems/sort-colors Title: 75. Sort Colors Problem Description: Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. You must solve this problem without using the library's sort function. Example 1: Input: nums = [2,0,2,1,1,0] Output: [0,0,1,1,2,2]

Example 2: Input: nums = [2,0,1] Output: [0,1,2]

Constraints:

n == nums.length 1 <= n <= 300 nums[i] is either 0, 1, or 2.

Follow up: Could you come up with a one-pass algorithm using only constant extra space?

===
Problem Number: 586 URL: https://leetcode.com/problems/combinations Title: 77. Combinations Problem Description: Given two integers n and k, return all possible combinations of k numbers chosen from the range [1, n]. You may return the answer in any order. Example 1: Input: n = 4, k = 2 Output: [[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]] Explanation: There are 4 choose 2 = 6 total combinations. Note that combinations are unordered, i.e., [1,2] and [2,1] are considered to be the same combination.

Example 2: Input: n = 1, k = 1 Output: [[1]] Explanation: There is 1 choose 1 = 1 total combination.

Constraints:

1 <= n <= 20 1 <= k <= n

===
Problem Number: 587 URL: https://leetcode.com/problems/subsets Title: 78. Subsets Problem Description: Given an integer array nums of unique elements, return all possible subsets (the power set). The solution set must not contain duplicate subsets. Return the solution in any order. Example 1: Input: nums = [1,2,3] Output: [[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]

Example 2: Input: nums = [0] Output: [[],[0]]

Constraints:

1 <= nums.length <= 10 -10 <= nums[i] <= 10 All the numbers of nums are unique.

===
Problem Number: 588 URL: https://leetcode.com/problems/word-search Title: 79. Word Search Problem Description: Given an m x n grid of characters board and a string word, return true if word exists in the grid. The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once. Example 1:

Input: board = [[”A”,”B”,”C”,”E”],[”S”,”F”,”C”,”S”],[”A”,”D”,”E”,”E”]], word = ”ABCCED” Output: true

Example 2:

Input: board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]], word = "SEE" Output: true

Example 3:

Input: board = [["A","B","C","E"],["S","F","C","S"],["A","D","E","E"]], word = "ABCB" Output: false

Constraints:

m == board.length n = board[i].length 1 <= m, n <= 6 1 <= word.length <= 15 board and word consists of only lowercase and uppercase English letters.

Follow up: Could you use search pruning to make your solution faster with a larger board?

==

Problem Number: 589 URL: https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii Title: 80. Remove Duplicates from Sorted Array II Problem Description: Given an integer array nums sorted in non-decreasing order, remove some duplicates in-place such that each unique element appears at most twice. The relative order of the elements should be kept the same. Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the first part of the array nums. More formally, if there are k elements after removing the duplicates, then the first k elements of nums should hold the final result. It does not matter what you leave beyond the first k elements. Return k after placing the final result in the first k slots of nums. Do not allocate extra space for another array. You must do this by modifying the input array in-place with O(1) extra memory. Custom Judge: The judge will test your solution with the following code: int[] nums = [...]; // Input array int[] expectedNums = [...]; // The expected answer with correct length

int k = removeDuplicates(nums); // Calls your implementation

assert k == expectedNums.length; for (int i = 0; i < k; i++) { assert nums[i] == expectedNums[i]; }

If all assertions pass, then your solution will be accepted. Example 1: Input: nums = [1,1,1,2,2,3] Output: 5, nums = [1,1,2,2,3,_] Explanation: Your function should return k = 5, with the first five elements of nums being 1, 1, 2, 2 and 3 respectively. It does not matter what you leave beyond the returned k (hence they are underscores).

Example 2: Input: nums = [0,0,1,1,1,1,2,3,3] Output: 7, nums = [0,0,1,1,2,3,3,_,_] Explanation: Your function should return k = 7, with the first seven elements of nums being 0, 0, 1, 1, 2, 3 and 3 respectively. It does not matter what you leave beyond the returned k (hence they are underscores).

Constraints:

1 <= nums.length <= 3 * 104 -104 <= nums[i] <= 104 nums is sorted in non-decreasing order.

===

Problem Number: 590 URL: https://leetcode.com/problems/search-in-rotated-sorted-array-ii Title: 81. Search in Rotated Sorted Array II Problem Description: There is an integer array nums sorted in non-decreasing order (not necessarily with distinct values). Before being passed to your function, nums is rotated at an unknown pivot index k (0 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,4,4,5,6,6,7] might be rotated at pivot index 5 and become [4,5,6,6,7,0,1,2,4,4]. Given the array nums after the rotation and an integer target, return true if target is in nums, or false if it is not in nums. You must decrease the overall operation steps as much as possible. Example 1: Input: nums = [2,5,6,0,0,1,2], target = 0 Output: true Example 2: Input: nums = [2,5,6,0,0,1,2], target = 3 Output: false

Constraints:

1 <= nums.length <= 5000 -104 <= nums[i] <= 104 nums is guaranteed to be rotated at some pivot. -104 <= target <= 104

Follow up: This problem is similar to Search in Rotated Sorted Array, but nums may contain duplicates. Would this affect the runtime complexity? How and why?

===

Problem Number: 591 URL: https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii Title: 82. Remove Duplicates from Sorted List II Problem Description: Given the head of a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list. Return the linked list sorted as well. Example 1:

Input: head = [1,2,3,3,4,4,5] Output: [1,2,5]

Example 2:

Input: head = [1,1,1,2,3] Output: [2,3]

Constraints:

The number of nodes in the list is in the range [0, 300]. -100 <= Node.val <= 100 The list is guaranteed to be sorted in ascending order.

===

Problem Number: 592 URL: https://leetcode.com/problems/partition-list Title: 86. Partition List Problem Description: Given the head of a linked list and a value x, partition it such that all nodes less than x come before nodes greater than or equal to x. You should preserve the original relative order of the nodes in each of the two partitions. Example 1:

Input: head = [1,4,3,2,5,2], x = 3 Output: [1,2,2,4,3,5]

Example 2: Input: head = [2,1], x = 2 Output: [1,2]

Constraints:

The number of nodes in the list is in the range [0, 200]. -100 <= Node.val <= 100 -200 <= x <= 200

==
Problem Number: 593 URL: https://leetcode.com/problems/gray-code Title: 89. Gray Code Problem Description: An n-bit gray code sequence is a sequence of 2n integers where:

Every integer is in the inclusive range [0, 2n - 1], The first integer is 0, An integer appears no more than once in the sequence, The binary representation of every pair of adjacent integers differs by exactly one bit, and The binary representation of the first and last integers differs by exactly one bit.

Given an integer n, return any valid n-bit gray code sequence. Example 1: Input: n = 2 Output: [0,1,3,2] Explanation: The binary representation of [0,1,3,2] is [00,01,11,10]. - 00 and 01 differ by one bit - 01 and 11 differ by one bit - 11 and 10 differ by one bit - 10 and 00 differ by one bit [0,2,3,1] is also a valid gray code sequence, whose binary representation is [00,10,11,01]. - 00 and 10 differ by one bit - 10 and 11 differ by one bit - 11 and 01 differ by one bit - 01 and 00 differ by one bit

Example 2: Input: n = 1 Output: [0,1]

Constraints:

1 <= n <= 16

==
Problem Number: 594 URL: https://leetcode.com/problems/subsets-ii Title: 90. Subsets II Problem Description: Given an integer array nums that may contain duplicates, return all possible subsets (the power set). The solution set must not contain duplicate subsets. Return the solution in any order. Example 1: Input: nums = [1,2,2] Output: [[],[1],[1,2],[1,2,2],[2],[2,2]] Example 2: Input: nums = [0] Output: [[],[0]]

Constraints:

1 <= nums.length <= 10 -10 <= nums[i] <= 10

==
Problem Number: 595 URL: https://leetcode.com/problems/decode-ways Title: 91. Decode Ways Problem Description: A message containing letters from A-Z can be encoded into numbers using the following mapping: 'A' -> "1" 'B' -> "2" ... 'Z' -> "26"

To decode an encoded message, all the digits must be grouped then mapped back into letters using the reverse of the mapping above (there may be multiple ways). For example, "11106" can be mapped into:

”AAJF” with the grouping (1 1 10 6) ”KJF” with the grouping (11 10 6)

Note that the grouping (1 11 06) is invalid because ”06” cannot be mapped into ’F’ since ”6” is different from ”06”. Given a string s containing only digits, return the number of ways to decode it. The test cases are generated so that the answer fits in a 32-bit integer. Example 1: Input: s = ”12” Output: 2 Explanation: ”12” could be decoded as ”AB” (1 2) or ”L” (12).

Example 2: Input: s = ”226” Output: 3 Explanation: ”226” could be decoded as ”BZ” (2 26), ”VF” (22 6), or ”BBF” (2 2 6).

Example 3: Input: s = ”06” Output: 0 Explanation: ”06” cannot be mapped to ”F” because of the leading zero (”6” is different from ”06”).

Constraints:

1 <= s.length <= 100 s contains only digits and may contain leading zero(s).

==

Problem Number: 596 URL: https://leetcode.com/problems/reverse-linked-list-ii Title: 92. Reverse Linked List II Problem Description: Given the head of a singly linked list and two integers left and right where left <= right, reverse the nodes of the list from position left to position right, and return the reversed list. Example 1:

Input: head = [1,2,3,4,5], left = 2, right = 4 Output: [1,4,3,2,5]

Example 2: Input: head = [5], left = 1, right = 1 Output: [5]

Constraints:

The number of nodes in the list is n. 1 <= n <= 500 -500 <= Node.val <= 500 1 <= left <= right <= n

Follow up: Could you do it in one pass? ==

Problem Number: 597 URL: https://leetcode.com/problems/restore-ip-addresses Title: 93. Restore IP Addresses Problem Description: A valid IP address consists of exactly four integers separated by single dots. Each integer is between 0 and 255 (inclusive) and cannot have leading zeros.

For example, ”0.1.2.201” and ”192.168.1.1” are valid IP addresses, but ”0.011.255.245”, ”192.168.1.312” and ”192.168@1.1” are invalid IP addresses.

Given a string s containing only digits, return all possible valid IP addresses that can be formed by inserting dots into s. You are not allowed to reorder or remove any digits in s. You may return the valid IP addresses in any order. Example 1: Input: s = ”25525511135” Output: [”255.255.11.135”,”255.255.111.35”]

Example 2: Input: s = ”0000” Output: [”0.0.0.0”]

Example 3: Input: s = ”101023” Output: [”1.0.10.23”,”1.0.102.3”,”10.1.0.23”,”10.10.2.3”,”101.0.2.3”]

Constraints:

1 <= s.length <= 20 s consists of digits only.

==
Problem Number: 598 URL: https://leetcode.com/problems/unique-binary-search-trees-ii Title: 95. Unique Binary Search Trees II Problem Description: Given an integer n, return all the structurally unique BST's (binary search trees), which has exactly n nodes of unique values from 1 to n. Return the answer in any order. Example 1:

Input: n = 3 Output: [[1,null,2,null,3],[1,null,3,2],[2,1,3],[3,1,null,null,2],[3,2,null,1]]

Example 2: Input: n = 1 Output: [[1]]

Constraints:

1 <= n <= 8

==
Problem Number: 599 URL: https://leetcode.com/problems/unique-binary-search-trees Title: 96. Unique Binary Search Trees Problem Description: Given an integer n, return the number of structurally unique BST's (binary search trees) which has exactly n nodes of unique values from 1 to n. Example 1:

Input: n = 3 Output: 5

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 19

==
Problem Number: 600 URL: https://leetcode.com/problems/interleaving-string Title: 97. Interleaving String Problem Description: Given strings s1, s2, and s3, find whether s3 is formed by an interleaving of s1 and s2. An interleaving of two strings s and t is a configuration where s and t are divided into n and m substrings respectively, such that:

s = s1 + s2 + ... + sn t = t1 + t2 + ... + tm |n - m| <= 1 The interleaving is s1 + t1 + s2 + t2 + s3 + t3 + ... or t1 + s1 + t2 + s2 + t3 + s3 + ...

Note: a + b is the concatenation of strings a and b. Example 1:

Input: s1 = "aabcc", s2 = "dbbca", s3 = "aadbbcbcac" Output: true Explanation: One way to obtain s3 is: Split s1 into s1 = "aa" + "bc" + "c", and s2 into s2 = "dbbc" + "a". Interleaving the two splits, we get "aa" + "dbbc" + "bc" + "a" + "c" = "aadbbcbcac". Since s3 can be obtained by interleaving s1 and s2, we return true.

Example 2: Input: s1 = "aabcc", s2 = "dbbca", s3 = "aadbbbaccc" Output: false Explanation: Notice how it is impossible to interleave s2 with any other string to obtain s3.

Example 3: Input: s1 = "", s2 = "", s3 = "" Output: true

Constraints:

0 <= s1.length, s2.length <= 100 0 <= s3.length <= 200 s1, s2, and s3 consist of lowercase English letters.

Follow up: Could you solve it using only O(s2.length) additional memory space?

==

Problem Number: 601 URL: https://leetcode.com/problems/validate-binary-search-tree Title: 98. Validate Binary Search Tree Problem Description: Given the root of a binary tree, determine if it is a valid binary search tree (BST). A valid BST is defined as follows:

The left subtree of a node contains only nodes with keys less than the node's key. The right subtree of a node contains only nodes with keys greater than the node's key. Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [2,1,3] Output: true

Example 2:

Input: root = [5,1,4,null,null,3,6] Output: false Explanation: The root node's value is 5 but its right child's value is 4.

Constraints:

The number of nodes in the tree is in the range [1, 104]. -231 <= Node.val <= 231 - 1

==

Problem Number: 602 URL: https://leetcode.com/problems/recover-binary-search-tree Title: 99. Recover Binary Search Tree Problem Description: You are given the root of a binary search tree (BST), where the values of exactly two nodes of the tree were swapped by mistake. Recover the tree without changing its structure. Example 1:

Input: root = [1,3,null,null,2] Output: [3,1,null,null,2] Explanation: 3 cannot be a left child of 1 because 3 > 1. Swapping 1 and 3 makes the BST valid.

Example 2:

Input: root = [3,1,4,null,null,2] Output: [2,1,4,null,null,3] Explanation: 2 cannot be in the right subtree of 3 because 2 < 3. Swapping 2 and 3 makes the BST valid.

Constraints:

The number of nodes in the tree is in the range [2, 1000]. -231 <= Node.val <= 231 - 1

Follow up: A solution using O(n) space is pretty straight-forward. Could you devise a constant O(1) space solution? == Problem Number: 603 URL: https://leetcode.com/problems/binary-tree-level-order-traversal Title: 102. Binary Tree Level Order Traversal Problem Description: Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level). Example 1:

Input: root = [3,9,20,null,null,15,7] Output: [[3],[9,20],[15,7]]

Example 2: Input: root = [1] Output: [[1]]

Example 3: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 2000]. -1000 <= Node.val <= 1000

== Problem Number: 604 URL: https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal Title: 103. Binary Tree Zigzag Level Order Traversal Problem Description: Given the root of a binary tree, return the zigzag level order traversal of its nodes' values. (i.e., from left to right, then right to left for the next level and alternate between). Example 1:

Input: root = [3,9,20,null,null,15,7] Output: [[3],[20,9],[15,7]]

Example 2: Input: root = [1] Output: [[1]]

Example 3: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 2000]. -100 <= Node.val <= 100

== Problem Number: 605 URL: https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal Title: 105. Construct Binary Tree from Preorder and Inorder Traversal Problem Description: Given two integer arrays preorder and inorder where preorder is the preorder traversal of a binary tree and inorder is the inorder traversal of the same tree, construct and return the binary tree. Example 1:

Input: preorder = [3,9,20,15,7], inorder = [9,3,15,20,7] Output: [3,9,20,null,null,15,7]

Example 2: Input: preorder = [-1], inorder = [-1] Output: [-1]

Constraints:

1 <= preorder.length <= 3000 inorder.length == preorder.length -3000 <= preorder[i], inorder[i] <= 3000 preorder and inorder consist of unique values. Each value of inorder also appears in preorder. preorder is guaranteed to be the

preorder traversal of the tree. inorder is guaranteed to be the inorder traversal of the tree.

== Problem Number: 606 URL: https://leetcode.com/problems/construct-binary-tree-from-inorder-and-postorder-traversal Title: 106. Construct Binary Tree from Inorder and Postorder Traversal Problem Description: Given two integer arrays inorder and postorder where inorder is the inorder traversal of a binary tree and postorder is the postorder traversal of the same tree, construct and return the binary tree. Example 1:

Input: inorder = [9,3,15,20,7], postorder = [9,15,7,20,3] Output: [3,9,20,null,null,15,7]

Example 2: Input: inorder = [-1], postorder = [-1] Output: [-1]

Constraints:

1 <= inorder.length <= 3000 postorder.length == inorder.length -3000 <= inorder[i], postorder[i] <= 3000 inorder and postorder consist of unique values. Each value of postorder also appears in inorder. inorder is guaranteed to be the inorder traversal of the tree. postorder is guaranteed to be the postorder traversal of the tree.

== Problem Number: 607 URL: https://leetcode.com/problems/binary-tree-level-order-traversal-ii Title: 107. Binary Tree Level Order Traversal II Problem Description: Given the root of a binary tree, return the bottom-up level order traversal of its nodes' values. (i.e., from left to right, level by level from leaf to root). Example 1:

Input: root = [3,9,20,null,null,15,7] Output: [[15,7],[9,20],[3]]

Example 2: Input: root = [1] Output: [[1]]

Example 3: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 2000]. -1000 <= Node.val <= 1000

== Problem Number: 608 URL: https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree Title: 109. Convert Sorted List to Binary Search Tree Problem Description: Given the head of a singly linked list where elements are sorted in ascending order, convert it to a height-balanced binary search tree. Example 1:

Input: head = [-10,-3,0,5,9] Output: [0,-3,9,-10,null,5] Explanation: One possible answer is [0,-3,9,-10,null,5], which represents the shown height balanced BST.

Example 2: Input: head = [] Output: []

Constraints:

The number of nodes in head is in the range [0, 2 * 104]. -105 <= Node.val <= 105

==

Problem Number: 609 URL: https://leetcode.com/problems/path-sum-ii Title: 113. Path Sum II Problem Description: Given the root of a binary tree and an integer targetSum, return all root-to-leaf paths where the sum of the node values in the path equals targetSum. Each path should be returned as a list of the node values, not node references. A root-to-leaf path is a path starting from the root and ending at any leaf node. A leaf is a node with no children. Example 1:

Input: root = [5,4,8,11,null,13,4,7,2,null,null,5,1], targetSum = 22 Output: [[5,4,11,2],[5,8,4,5]] Explanation: There are two paths whose sum equals targetSum: 5 + 4 + 11 + 2 = 22 5 + 8 + 4 + 5 = 22

Example 2:

Input: root = [1,2,3], targetSum = 5 Output: []

Example 3: Input: root = [1,2], targetSum = 0 Output: []

Constraints:

The number of nodes in the tree is in the range [0, 5000]. -1000 <= Node.val <= 1000 -1000 <= targetSum <= 1000

==

Problem Number: 610 URL: https://leetcode.com/problems/flatten-binary-tree-to-linked-list Title: 114. Flatten Binary Tree to Linked List Problem Description: Given the root of a binary tree, flatten the tree into a "linked list":

The "linked list" should use the same TreeNode class where the right child pointer points to the next node in the list and the left child pointer is always null. The "linked list" should be in the same order as a pre-order traversal of the binary tree.

Example 1:

Input: root = [1,2,5,3,4,null,6] Output: [1,null,2,null,3,null,4,null,5,null,6]

Example 2: Input: root = [] Output: []

Example 3: Input: root = [0] Output: [0]

Constraints:

The number of nodes in the tree is in the range [0, 2000]. -100 <= Node.val <= 100

Follow up: Can you flatten the tree in-place (with O(1) extra space)? ==

Problem Number: 611 URL: https://leetcode.com/problems/populating-next-right-pointers-in-each-node Title: 116. Populating Next Right Pointers in Each Node Problem Description: You are given a perfect binary tree where all leaves are on the same level, and every parent has two children. The binary tree has the following definition: struct Node { int val; Node *left; Node *right; Node *next; }

Populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to NULL. Initially, all next pointers are set to NULL.   Example 1:

Input: root = [1,2,3,4,5,6,7] Output: [1,#,2,3,#,4,5,6,7,#] Explanation: Given the above perfect binary tree (Figure A), your function should populate each next pointer to point to its next right node, just like in Figure B. The serialized output is in level order as connected by the next pointers, with '#' signifying the end of each level.

Example 2: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 212 - 1]. -1000 <= Node.val <= 1000

Follow-up:

You may only use constant extra space. The recursive approach is fine. You may assume implicit stack space does not count as extra space for this problem.

==

Problem Number: 612 URL: https://leetcode.com/problems/populating-next-right-pointers-in-each-node-ii Title: 117. Populating Next Right Pointers in Each Node II Problem Description: Given a binary tree struct Node { int val; Node *left; Node *right; Node *next; }

Populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to NULL. Initially, all next pointers are set to NULL.   Example 1:

Input: root = [1,2,3,4,5,null,7] Output: [1,#,2,3,#,4,5,7,#] Explanation: Given the above binary tree (Figure A), your function should populate each next pointer to point to its next right node, just like in Figure B. The serialized output is in level order as connected by the next pointers, with '#' signifying the end of each level.

Example 2: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 6000]. -100 <= Node.val <= 100

Follow-up:

You may only use constant extra space. The recursive approach is fine. You may assume implicit stack space does not count as extra space for this problem.

==
Problem Number: 613 URL: https://leetcode.com/problems/triangle Title: 120. Triangle Problem Description: Given a triangle array, return the minimum path sum from top to bottom. For each step, you may move to an adjacent number of the row below. More formally, if you are on index i on the current row, you may move to either index i or index i + 1 on the next row. Example 1: Input: triangle = [[2],[3,4],[6,5,7],[4,1,8,3]] Output: 11 Explanation: The triangle looks like: 2 3 4 6 5 7 4 1 8 3 The minimum path sum from top to bottom is 2 + 3 + 5 + 1 = 11 (underlined above).

Example 2: Input: triangle = [[-10]] Output: -10

Constraints:

1 <= triangle.length <= 200 triangle[0].length == 1 triangle[i].length == triangle[i - 1].length + 1 -104 <= triangle[i][j] <= 104

Follow up: Could you do this using only O(n) extra space, where n is the total number of rows in the triangle? ==
Problem Number: 614 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii Title: 122. Best Time to Buy and Sell Stock II Problem Description: You are given an integer array prices where prices[i] is the price of a given stock on the ith day. On each day, you may decide to buy and/or sell the stock. You can only hold at most one share of the stock at any time. However, you can buy it then immediately sell it on the same day. Find and return the maximum profit you can achieve. Example 1: Input: prices = [7,1,5,3,6,4] Output: 7 Explanation: Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4. Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3. Total profit is 4 + 3 = 7.

Example 2: Input: prices = [1,2,3,4,5] Output: 4 Explanation: Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4. Total profit is 4.

Example 3: Input: prices = [7,6,4,3,1] Output: 0 Explanation: There is no way to make a positive profit, so we never buy the stock to achieve the maximum profit of 0.

Constraints:

1 <= prices.length <= 3 * 104 0 <= prices[i] <= 104

==
Problem Number: 615 URL: https://leetcode.com/problems/longest-consecutive-sequence Title: 128. Longest Consecutive Sequence Problem

Description: Given an unsorted array of integers nums, return the length of the longest consecutive elements sequence. You must write an algorithm that runs in O(n) time. Example 1: Input: nums = [100,4,200,1,3,2] Output: 4 Explanation: The longest consecutive elements sequence is [1, 2, 3, 4]. Therefore its length is 4.

Example 2: Input: nums = [0,3,7,2,5,8,4,6,0,1] Output: 9

Constraints:

0 <= nums.length <= 105 -109 <= nums[i] <= 109

==

Problem Number: 616 URL: https://leetcode.com/problems/sum-root-to-leaf-numbers Title: 129. Sum Root to Leaf Numbers Problem Description: You are given the root of a binary tree containing digits from 0 to 9 only. Each root-to-leaf path in the tree represents a number.

For example, the root-to-leaf path 1 -> 2 -> 3 represents the number 123.

Return the total sum of all root-to-leaf numbers. Test cases are generated so that the answer will fit in a 32-bit integer. A leaf node is a node with no children. Example 1:

Input: root = [1,2,3] Output: 25 Explanation: The root-to-leaf path 1->2 represents the number 12. The root-to-leaf path 1->3 represents the number 13. Therefore, sum = 12 + 13 = 25.

Example 2:

Input: root = [4,9,0,5,1] Output: 1026 Explanation: The root-to-leaf path 4->9->5 represents the number 495. The root-to-leaf path 4->9->1 represents the number 491. The root-to-leaf path 4->0 represents the number 40. Therefore, sum = 495 + 491 + 40 = 1026.

Constraints:

The number of nodes in the tree is in the range [1, 1000]. 0 <= Node.val <= 9 The depth of the tree will not exceed 10.

==

Problem Number: 617 URL: https://leetcode.com/problems/surrounded-regions Title: 130. Surrounded Regions Problem Description: Given an m x n matrix board containing ’X’ and ’O’, capture all regions that are 4-directionally surrounded by ’X’. A region is captured by flipping all ’O’s into ’X’s in that surrounded region. Example 1:

Input: board = [[”X”,”X”,”X”,”X”],[”X”,”O”,”O”,”X”],[”X”,”X”,”O”,”X”],[”X”,”O”,”X”,”X”]] Output: [[”X”,”X”,”X”,”X”],[”X”,”X”,”X”,”X”],[”X”,”X”,”X”,”X”],[”X”,”O”,”X”,”X”]] Explanation: Notice that an ’O’ should not be flipped if: - It is on the border, or - It is adjacent to an ’O’ that should not be flipped. The bottom ’O’ is on the border, so it is not flipped. The other three 'O' form a surrounded region, so they are flipped.

Example 2: Input: board = [["X"]] Output: [["X"]]

Constraints:

m == board.length n == board[i].length 1 <= m, n <= 200 board[i][j] is 'X' or 'O'.

==
Problem Number: 618 URL: https://leetcode.com/problems/palindrome-partitioning Title: 131. Palindrome Partitioning Problem Description: Given a string s, partition s such that every substring of the partition is a palindrome. Return all possible palindrome partitioning of s. Example 1: Input: s = "aab" Output: [["a","a","b"],["aa","b"]] Example 2: Input: s = "a" Output: [["a"]]

Constraints:

1 <= s.length <= 16 s contains only lowercase English letters.

==
Problem Number: 619 URL: https://leetcode.com/problems/clone-graph Title: 133. Clone Graph Problem Description: Given a reference of a node in a connected undirected graph. Return a deep copy (clone) of the graph. Each node in the graph contains a value (int) and a list (List[Node]) of its neighbors. class Node { public int val; public List<Node> neighbors; }

Test case format: For simplicity, each node's value is the same as the node's index (1-indexed). For example, the first node with val == 1, the second node with val == 2, and so on. The graph is represented in the test case using an adjacency list. An adjacency list is a collection of unordered lists used to represent a finite graph. Each list describes the set of neighbors of a node in the graph. The given node will always be the first node with val = 1. You must return the copy of the given node as a reference to the cloned graph. Example 1:

Input: adjList = [[2,4],[1,3],[2,4],[1,3]] Output: [[2,4],[1,3],[2,4],[1,3]] Explanation: There are 4 nodes in the graph. 1st node (val = 1)'s neighbors are 2nd node (val = 2) and 4th node (val = 4). 2nd node (val = 2)'s neighbors are 1st node (val = 1) and 3rd node (val = 3). 3rd node (val = 3)'s neighbors are 2nd node (val = 2) and 4th node (val = 4). 4th node (val = 4)'s neighbors are 1st node (val = 1) and 3rd node (val = 3).

Example 2:

Input: adjList = [[]] Output: [[]] Explanation: Note that the input contains one empty list. The graph consists of only one node with val = 1 and it does not have any neighbors.

Example 3: Input: adjList = [] Output: [] Explanation: This an empty graph, it does not have any nodes.

Constraints:

The number of nodes in the graph is in the range [0, 100]. 1 <= Node.val <= 100 Node.val is unique for each node. There are no repeated edges and no self-loops in the graph. The Graph is connected and all nodes can be visited starting from the given node.

==

Problem Number: 620 URL: https://leetcode.com/problems/gas-station Title: 134. Gas Station Problem Description: There are n gas stations along a circular route, where the amount of gas at the ith station is gas[i]. You have a car with an unlimited gas tank and it costs cost[i] of gas to travel from the ith station to its next (i + 1)th station. You begin the journey with an empty tank at one of the gas stations. Given two integer arrays gas and cost, return the starting gas station's index if you can travel around the circuit once in the clockwise direction, otherwise return -1. If there exists a solution, it is guaranteed to be unique Example 1: Input: gas = [1,2,3,4,5], cost = [3,4,5,1,2] Output: 3 Explanation: Start at station 3 (index 3) and fill up with 4 unit of gas. Your tank = 0 + 4 = 4 Travel to station 4. Your tank = 4 - 1 + 5 = 8 Travel to station 0. Your tank = 8 - 2 + 1 = 7 Travel to station 1. Your tank = 7 - 3 + 2 = 6 Travel to station 2. Your tank = 6 - 4 + 3 = 5 Travel to station 3. The cost is 5. Your gas is just enough to travel back to station 3. Therefore, return 3 as the starting index.

Example 2: Input: gas = [2,3,4], cost = [3,4,3] Output: -1 Explanation: You can't start at station 0 or 1, as there is not enough gas to travel to the next station. Let's start at station 2 and fill up with 4 unit of gas. Your tank = 0 + 4 = 4 Travel to station 0. Your tank = 4 - 3 + 2 = 3 Travel to station 1. Your tank = 3 - 3 + 3 = 3 You cannot travel back to station 2, as it requires 4 unit of gas but you only have 3. Therefore, you can't travel around the circuit once no matter where you start.

Constraints:

n == gas.length == cost.length 1 <= n <= 105 0 <= gas[i], cost[i] <= 104

==

Problem Number: 621 URL: https://leetcode.com/problems/single-number-ii Title: 137. Single Number II Problem Description: Given an integer array nums where every element appears three times except for one, which appears exactly once. Find the single element and return it. You must implement a solution with a linear runtime complexity and use only constant extra space. Example 1: Input: nums = [2,2,3,2] Output: 3 Example 2: Input: nums = [0,1,0,1,0,1,99] Output: 99

Constraints:

1 <= nums.length <= 3 * 104 -231 <= nums[i] <= 231 - 1 Each element in nums appears exactly three times except for one element which appears once.

==

Problem Number: 622 URL: https://leetcode.com/problems/copy-list-with-random-pointer Title: 138. Copy List with Random Pointer Problem Description: A linked list of length n is given such that each node contains an additional random pointer, which could point to any node in the list, or null. Construct a deep copy of the list. The deep copy should consist of exactly n brand new nodes, where each new node has its value set to the value of its corresponding original node. Both the next and random pointer of the new nodes should point to new nodes in the copied list such that the pointers in the original list and copied list represent the same list state. None of the pointers in the new list should point to nodes in the original list. For example, if there are two nodes X and Y in the original list, where X.random --> Y, then for the corresponding two nodes x and y in the copied list, x.random --> y. Return the head of the copied linked list. The linked list is represented in the input/output as a list of n nodes. Each node is represented as a pair of [val, random_index] where:

val: an integer representing Node.val random_index: the index of the node (range from 0 to n-1) that the random pointer points to, or null if it does not point to any node.

Your code will only be given the head of the original linked list.   Example 1:

Input: head = [[7,null],[13,0],[11,4],[10,2],[1,0]] Output: [[7,null],[13,0],[11,4],[10,2],[1,0]]

Example 2:

Input: head = [[1,1],[2,1]] Output: [[1,1],[2,1]]

Example 3:

Input: head = [[3,null],[3,0],[3,null]] Output: [[3,null],[3,0],[3,null]]

Constraints:

0 <= n <= 1000 -104 <= Node.val <= 104 Node.random is null or is pointing to some node in the linked list.

==

Problem Number: 623 URL: https://leetcode.com/problems/word-break Title: 139. Word Break Problem Description: Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words. Note that the same word in the dictionary may be reused multiple times in the segmentation.   Example 1: Input: s = "leetcode", wordDict = ["leet","code"] Output: true Explanation: Return true because "leetcode" can be segmented as "leet code".

Example 2: Input: s = "applepenapple", wordDict = ["apple","pen"] Output: true Explanation: Return true because "applepenapple" can be segmented as "apple pen apple". Note that you are allowed to reuse a dictionary word.

Example 3: Input: s = "catsandog", wordDict = ["cats","dog","sand","and","cat"] Output: false

Constraints:

1 <= s.length <= 300 1 <= wordDict.length <= 1000 1 <= wordDict[i].length <= 20 s and wordDict[i] consist of only lowercase English letters. All the strings of wordDict are unique.

== Problem Number: 624 URL: https://leetcode.com/problems/linked-list-cycle-ii Title: 142. Linked List Cycle II Problem Description: Given the head of a linked list, return the node where the cycle begins. If there is no cycle, return null. There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to (0-indexed). It is -1 if there is no cycle. Note that pos is not passed as a parameter. Do not modify the linked list. Example 1:

Input: head = [3,2,0,-4], pos = 1 Output: tail connects to node index 1 Explanation: There is a cycle in the linked list, where tail connects to the second node.

Example 2:

Input: head = [1,2], pos = 0 Output: tail connects to node index 0 Explanation: There is a cycle in the linked list, where tail connects to the first node.

Example 3:

Input: head = [1], pos = -1 Output: no cycle Explanation: There is no cycle in the linked list.

Constraints:

The number of the nodes in the list is in the range [0, 104]. -105 <= Node.val <= 105 pos is -1 or a valid index in the linked-list.

Follow up: Can you solve it using O(1) (i.e. constant) memory?

== Problem Number: 625 URL: https://leetcode.com/problems/reorder-list Title: 143. Reorder List Problem Description: You are given the head of a singly linked-list. The list can be represented as: L0 → L1 → … → Ln - 1 → Ln

Reorder the list to be on the following form: L0 → Ln → L1 → Ln - 1 → L2 → Ln - 2 → …

You may not modify the values in the list's nodes. Only nodes themselves may be changed. Example 1:

Input: head = [1,2,3,4] Output: [1,4,2,3]

Example 2:

Input: head = [1,2,3,4,5] Output: [1,5,2,4,3]

Constraints:

The number of nodes in the list is in the range [1, 5 * 104]. 1 <= Node.val <= 1000

==
Problem Number: 626 URL: https://leetcode.com/problems/lru-cache Title: 146. LRU Cache Problem Description: Design a data structure that follows the constraints of a Least Recently Used (LRU) cache. Implement the LRUCache class:

LRUCache(int capacity) Initialize the LRU cache with positive size capacity. int get(int key) Return the value of the key if the key exists, otherwise return -1. void put(int key, int value) Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, evict the least recently used key.

The functions get and put must each run in O(1) average time complexity. Example 1: Input ["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"] [[2], [1, 1], [2, 2], [1], [3, 3], [2], [4, 4], [1], [3], [4]] Output [null, null, null, 1, null, -1, null, -1, 3, 4]

Explanation LRUCache lRUCache = new LRUCache(2); lRUCache.put(1, 1); // cache is {1=1} lRUCache.put(2, 2); // cache is {1=1, 2=2} lRUCache.get(1); // return 1 lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3} lRUCache.get(2); // returns -1 (not found) lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3} lRUCache.get(1); // return -1 (not found) lRUCache.get(3); // return 3 lRUCache.get(4); // return 4

Constraints:

1 <= capacity <= 3000 0 <= key <= 104 0 <= value <= 105 At most 2 * 105 calls will be made to get and put.

==
Problem Number: 627 URL: https://leetcode.com/problems/insertion-sort-list Title: 147. Insertion Sort List Problem Description: Given the head of a singly linked list, sort the list using insertion sort, and return the sorted list's head. The steps of the insertion sort algorithm:

Insertion sort iterates, consuming one input element each repetition and growing a sorted output list. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list and inserts it there. It repeats until no input elements remain.

The following is a graphical example of the insertion sort algorithm. The partially sorted list (black) initially contains only the first element in the list. One element (red) is removed from the input data and inserted in-place into the sorted list with each iteration.

Example 1:

Input: head = [4,2,1,3] Output: [1,2,3,4]

Example 2:

Input: head = [-1,5,3,4,0] Output: [-1,0,3,4,5]

Constraints:

The number of nodes in the list is in the range [1, 5000]. -5000 <= Node.val <= 5000

==
Problem Number: 628 URL: https://leetcode.com/problems/sort-list Title: 148. Sort List Problem Description: Given the head of a linked list, return the list after sorting it in ascending order. Example 1:

Input: head = [4,2,1,3] Output: [1,2,3,4]

Example 2:

Input: head = [-1,5,3,4,0] Output: [-1,0,3,4,5]

Example 3: Input: head = [] Output: []

Constraints:

The number of nodes in the list is in the range [0, 5 * 104]. -105 <= Node.val <= 105

Follow up: Can you sort the linked list in O(n logn) time and O(1) memory (i.e. constant space)?

==
Problem Number: 629 URL: https://leetcode.com/problems/evaluate-reverse-polish-notation Title: 150. Evaluate Reverse Polish Notation Problem Description: You are given an array of strings tokens that represents an arithmetic expression in a Reverse Polish Notation. Evaluate the expression. Return an integer that represents the value of the expression. Note that:

The valid operators are '+', '-', '*', and '/'. Each operand may be an integer or another expression. The division between two integers always truncates toward zero. There will not be any division by zero. The input represents a valid arithmetic expression in a reverse polish notation. The answer and all the intermediate calculations can be represented in a 32-bit integer.

Example 1: Input: tokens = ["2","1","+","3","*"] Output: 9 Explanation: ((2 + 1) * 3) = 9

Example 2: Input: tokens = ["4","13","5","/","+"] Output: 6 Explanation: (4 + (13 / 5)) = 6

Example 3: Input: tokens = ["10","6","9","3","+","-11","*","/","*","17","+","5","+"] Output: 22 Explanation: ((10 * (6 / ((9 + 3) * -11))) + 17) + 5 = ((10 * (6 /

(12 * -11))) + 17) + 5 = ((10 * (6 / -132)) + 17) + 5 = ((10 * 0) + 17) + 5 = (0 + 17) + 5 = 17 + 5 = 22

Constraints:

1 <= tokens.length <= 104 tokens[i] is either an operator: ”+”, ”-”, ”*”, or ”/”, or an integer in the range [-200, 200].

==
Problem Number: 630 URL: https://leetcode.com/problems/reverse-words-in-a-string Title: 151. Reverse Words in a String Problem Description: Given an input string s, reverse the order of the words. A word is defined as a sequence of non-space characters. The words in s will be separated by at least one space. Return a string of the words in reverse order concatenated by a single space. Note that s may contain leading or trailing spaces or multiple spaces between two words. The returned string should only have a single space separating the words. Do not include any extra spaces. Example 1: Input: s = ”the sky is blue” Output: ”blue is sky the”

Example 2: Input: s = ” hello world ” Output: ”world hello” Explanation: Your reversed string should not contain leading or trailing spaces.

Example 3: Input: s = ”a good example” Output: ”example good a” Explanation: You need to reduce multiple spaces between two words to a single space in the reversed string.

Constraints:

1 <= s.length <= 104 s contains English letters (upper-case and lower-case), digits, and spaces ’ ’. There is at least one word in s.

Follow-up: If the string data type is mutable in your language, can you solve it in-place with O(1) extra space?

==
Problem Number: 631 URL: https://leetcode.com/problems/maximum-product-subarray Title: 152. Maximum Product Subarray Problem Description: Given an integer array nums, find a subarray that has the largest product, and return the product. The test cases are generated so that the answer will fit in a 32-bit integer. Example 1: Input: nums = [2,3,-2,4] Output: 6 Explanation: [2,3] has the largest product 6.

Example 2: Input: nums = [-2,0,-1] Output: 0 Explanation: The result cannot be 2, because [-2,-1] is not a subarray.

Constraints:

1 <= nums.length <= 2 * 104 -10 <= nums[i] <= 10 The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

==
Problem Number: 632 URL: https://leetcode.com/problems/find-minimum-

in-rotated-sorted-array Title: 153. Find Minimum in Rotated Sorted Array Problem Description: Suppose an array of length n sorted in ascending order is rotated between 1 and n times. For example, the array nums = [0,1,2,4,5,6,7] might become:

[4,5,6,7,0,1,2] if it was rotated 4 times. [0,1,2,4,5,6,7] if it was rotated 7 times.

Notice that rotating an array [a[0], a[1], a[2], ..., a[n-1]] 1 time results in the array [a[n-1], a[0], a[1], a[2], ..., a[n-2]]. Given the sorted rotated array nums of unique elements, return the minimum element of this array. You must write an algorithm that runs in O(log n) time.   Example 1: Input: nums = [3,4,5,1,2] Output: 1 Explanation: The original array was [1,2,3,4,5] rotated 3 times.

Example 2: Input: nums = [4,5,6,7,0,1,2] Output: 0 Explanation: The original array was [0,1,2,4,5,6,7] and it was rotated 4 times.

Example 3: Input: nums = [11,13,15,17] Output: 11 Explanation: The original array was [11,13,15,17] and it was rotated 4 times.

Constraints:

n == nums.length 1 <= n <= 5000 -5000 <= nums[i] <= 5000 All the integers of nums are unique. nums is sorted and rotated between 1 and n times.

==
Problem Number: 633 URL: https://leetcode.com/problems/min-stack Title: 155. Min Stack Problem Description: Design a stack that supports push, pop, top, and retrieving the minimum element in constant time. Implement the MinStack class:

MinStack() initializes the stack object. void push(int val) pushes the element val onto the stack. void pop() removes the element on the top of the stack. int top() gets the top element of the stack. int getMin() retrieves the minimum element in the stack.

You must implement a solution with O(1) time complexity for each function. Example 1: Input ["MinStack","push","push","push","getMin","pop","top","getMin"] [[],[-2],[0],[-3],[],[],[],[]]

Output [null,null,null,null,-3,null,0,-2]

Explanation MinStack minStack = new MinStack(); minStack.push(-2); minStack.push(0); minStack.push(-3); minStack.getMin(); // return -3 minStack.pop(); minStack.top(); // return 0 minStack.getMin(); // return -2

Constraints:

-231 <= val <= 231 - 1 Methods pop, top and getMin operations will always be called on non-empty stacks. At most 3 * 104 calls will be made to push, pop, top, and getMin.

==
Problem Number: 634 URL: https://leetcode.com/problems/find-peak-element

Title: 162. Find Peak Element Problem Description: A peak element is an element that is strictly greater than its neighbors. Given a 0-indexed integer array nums, find a peak element, and return its index. If the array contains multiple peaks, return the index to any of the peaks. You may imagine that nums[-1] = nums[n] = -∞. In other words, an element is always considered to be strictly greater than a neighbor that is outside the array. You must write an algorithm that runs in O(log n) time. Example 1: Input: nums = [1,2,3,1] Output: 2 Explanation: 3 is a peak element and your function should return the index number 2. Example 2: Input: nums = [1,2,1,3,5,6,4] Output: 5 Explanation: Your function can return either index number 1 where the peak element is 2, or index number 5 where the peak element is 6. Constraints:

1 <= nums.length <= 1000 -231 <= nums[i] <= 231 - 1 nums[i] != nums[i + 1] for all valid i.

==
Problem Number: 635 URL: https://leetcode.com/problems/compare-version-numbers Title: 165. Compare Version Numbers Problem Description: Given two version numbers, version1 and version2, compare them.

Version numbers consist of one or more revisions joined by a dot '.'. Each revision consists of digits and may contain leading zeros. Every revision contains at least one character. Revisions are 0-indexed from left to right, with the leftmost revision being revision 0, the next revision being revision 1, and so on. For example 2.5.33 and 0.1 are valid version numbers. To compare version numbers, compare their revisions in left-to-right order. Revisions are compared using their integer value ignoring any leading zeros. This means that revisions 1 and 001 are considered equal. If a version number does not specify a revision at an index, then treat the revision as 0. For example, version 1.0 is less than version 1.1 because their revision 0s are the same, but their revision 1s are 0 and 1 respectively, and 0 < 1. Return the following:

If version1 < version2, return -1. If version1 > version2, return 1. Otherwise, return 0.

Example 1: Input: version1 = "1.01", version2 = "1.001" Output: 0 Explanation: Ignoring leading zeroes, both "01" and "001" represent the same integer "1".

Example 2: Input: version1 = "1.0", version2 = "1.0.0" Output: 0 Explanation: version1 does not specify revision 2, which means it is treated as "0".

Example 3: Input: version1 = "0.1", version2 = "1.1" Output: -1 Explanation: version1's revision 0 is "0", while version2's revision 0 is "1". 0 < 1, so version1 < version2.

Constraints:

1 <= version1.length, version2.length <= 500 version1 and version2 only contain digits and '.'. version1 and version2 are valid version numbers. All the given

revisions in version1 and version2 can be stored in a 32-bit integer.

== Problem Number: 636 URL: https://leetcode.com/problems/fraction-to-recurring-decimal Title: 166. Fraction to Recurring Decimal Problem Description: Given two integers representing the numerator and denominator of a fraction, return the fraction in string format. If the fractional part is repeating, enclose the repeating part in parentheses. If multiple answers are possible, return any of them. It is guaranteed that the length of the answer string is less than 104 for all the given inputs. Example 1: Input: numerator = 1, denominator = 2 Output: "0.5"

Example 2: Input: numerator = 2, denominator = 1 Output: "2"

Example 3: Input: numerator = 4, denominator = 333 Output: "0.(012)"

Constraints:

-231 <= numerator, denominator <= 231 - 1 denominator != 0

== Problem Number: 637 URL: https://leetcode.com/problems/two-sum-ii-input-array-is-sorted Title: 167. Two Sum II - Input Array Is Sorted Problem Description: Given a 1-indexed array of integers numbers that is already sorted in non-decreasing order, find two numbers such that they add up to a specific target number. Let these two numbers be numbers[index1] and numbers[index2] where 1 <= index1 < index2 < numbers.length. Return the indices of the two numbers, index1 and index2, added by one as an integer array [index1, index2] of length 2. The tests are generated such that there is exactly one solution. You may not use the same element twice. Your solution must use only constant extra space. Example 1: Input: numbers = [2,7,11,15], target = 9 Output: [1,2] Explanation: The sum of 2 and 7 is 9. Therefore, index1 = 1, index2 = 2. We return [1, 2].

Example 2: Input: numbers = [2,3,4], target = 6 Output: [1,3] Explanation: The sum of 2 and 4 is 6. Therefore index1 = 1, index2 = 3. We return [1, 3].

Example 3: Input: numbers = [-1,0], target = -1 Output: [1,2] Explanation: The sum of -1 and 0 is -1. Therefore index1 = 1, index2 = 2. We return [1, 2].

Constraints:

2 <= numbers.length <= 3 * 104 -1000 <= numbers[i] <= 1000 numbers is sorted in non-decreasing order. -1000 <= target <= 1000 The tests are generated such that there is exactly one solution.

== Problem Number: 638 URL: https://leetcode.com/problems/factorial-trailing-zeroes Title: 172. Factorial Trailing Zeroes Problem Description: Given an integer n, return the number of trailing zeroes in n!. Note that n! = n * (n - 1)

* (n - 2) * ... * 3 * 2 * 1. Example 1: Input: n = 3 Output: 0 Explanation: 3! = 6, no trailing zero.

Example 2: Input: n = 5 Output: 1 Explanation: 5! = 120, one trailing zero.

Example 3: Input: n = 0 Output: 0

Constraints:

0 <= n <= 104

Follow up: Could you write a solution that works in logarithmic time complexity?

==
Problem Number: 639 URL: https://leetcode.com/problems/binary-search-tree-iterator Title: 173. Binary Search Tree Iterator Problem Description: Implement the BSTIterator class that represents an iterator over the in-order traversal of a binary search tree (BST):

BSTIterator(TreeNode root) Initializes an object of the BSTIterator class. The root of the BST is given as part of the constructor. The pointer should be initialized to a non-existent number smaller than any element in the BST. boolean hasNext() Returns true if there exists a number in the traversal to the right of the pointer, otherwise returns false. int next() Moves the pointer to the right, then returns the number at the pointer.

Notice that by initializing the pointer to a non-existent smallest number, the first call to next() will return the smallest element in the BST. You may assume that next() calls will always be valid. That is, there will be at least a next number in the in-order traversal when next() is called. Example 1:

Input ["BSTIterator", "next", "next", "hasNext", "next", "hasNext", "next", "hasNext", "next", "hasNext"] [[[7, 3, 15, null, null, 9, 20]], [], [], [], [], [], [], [], [], []] Output [null, 3, 7, true, 9, true, 15, true, 20, false]

Explanation BSTIterator bSTIterator = new BSTIterator([7, 3, 15, null, null, 9, 20]); bSTIterator.next(); // return 3 bSTIterator.next(); // return 7 bSTIterator.hasNext(); // return True bSTIterator.next(); // return 9 bSTIterator.hasNext(); // return True bSTIterator.next(); // return 15 bSTIterator.hasNext(); // return True bSTIterator.next(); // return 20 bSTIterator.hasNext(); // return False

Constraints:

The number of nodes in the tree is in the range [1, 105]. 0 <= Node.val <= 106 At most 105 calls will be made to hasNext, and next.

Follow up:

Could you implement next() and hasNext() to run in average O(1) time and use O(h) memory, where h is the height of the tree?

==
Problem Number: 640 URL: https://leetcode.com/problems/largest-number Title: 179. Largest Number Problem Description: Given a list of non-negative integers nums, arrange them such that they form the largest number and return it. Since the result may be very large, so you need to return a string instead of an integer. Example 1: Input: nums = [10,2] Output: ”210”

Example 2: Input: nums = [3,30,34,5,9] Output: ”9534330”

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 109
==
Problem Number: 641 URL: https://leetcode.com/problems/repeated-dna-sequences Title: 187. Repeated DNA Sequences Problem Description: The DNA sequence is composed of a series of nucleotides abbreviated as ’A’, ’C’, ’G’, and ’T’.

For example, ”ACGAATTCCG” is a DNA sequence.

When studying DNA, it is useful to identify repeated sequences within the DNA. Given a string s that represents a DNA sequence, return all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in any order. Example 1: Input: s = ”AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT” Output: [”AAAAACCCCC”,”CCCCCAAAAA”] Example 2: Input: s = ”AAAAAAAAAAAAA” Output: [”AAAAAAAAAA”]

Constraints:

1 <= s.length <= 105 s[i] is either ’A’, ’C’, ’G’, or ’T’.
==
Problem Number: 642 URL: https://leetcode.com/problems/rotate-array Title: 189. Rotate Array Problem Description: Given an integer array nums, rotate the array to the right by k steps, where k is non-negative. Example 1: Input: nums = [1,2,3,4,5,6,7], k = 3 Output: [5,6,7,1,2,3,4] Explanation: rotate 1 steps to the right: [7,1,2,3,4,5,6] rotate 2 steps to the right: [6,7,1,2,3,4,5] rotate 3 steps to the right: [5,6,7,1,2,3,4]

Example 2: Input: nums = [-1,-100,3,99], k = 2 Output: [3,99,-1,-100] Explanation: rotate 1 steps to the right: [99,-1,-100,3] rotate 2 steps to the right: [3,99,-1,-100]

Constraints:

1 <= nums.length <= 105 -231 <= nums[i] <= 231 - 1 0 <= k <= 105

Follow up:

Try to come up with as many solutions as you can. There are at least three different ways to solve this problem. Could you do it in-place with O(1) extra space?

==

Problem Number: 643 URL: https://leetcode.com/problems/house-robber Title: 198. House Robber Problem Description: You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and it will automatically contact the police if two adjacent houses were broken into on the same night. Given an integer array nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police. Example 1: Input: nums = [1,2,3,1] Output: 4 Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3). Total amount you can rob = 1 + 3 = 4.

Example 2: Input: nums = [2,7,9,3,1] Output: 12 Explanation: Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money = 1). Total amount you can rob = 2 + 9 + 1 = 12.

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 400

==

Problem Number: 644 URL: https://leetcode.com/problems/binary-tree-right-side-view Title: 199. Binary Tree Right Side View Problem Description: Given the root of a binary tree, imagine yourself standing on the right side of it, return the values of the nodes you can see ordered from top to bottom. Example 1:

Input: root = [1,2,3,null,5,null,4] Output: [1,3,4]

Example 2: Input: root = [1,null,3] Output: [1,3]

Example 3: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 100]. -100 <= Node.val <= 100

==

Problem Number: 645 URL: https://leetcode.com/problems/number-of-islands Title: 200. Number of Islands Problem Description: Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water. Example 1: Input: grid = [["1","1","1","1","0"], ["1","1","0","1","0"], ["1","1","0","0","0"], ["0","0","0","0","0"]] Output: 1

Example 2: Input: grid = [["1","1","0","0","0"], ["1","1","0","0","0"], ["0","0","1","0","0"], ["0","0","0","1","1"]] Output: 3

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 300 grid[i][j] is '0' or '1'.

==
Problem Number: 646 URL: https://leetcode.com/problems/bitwise-and-of-numbers-range Title: 201. Bitwise AND of Numbers Range Problem Description: Given two integers left and right that represent the range [left, right], return the bitwise AND of all numbers in this range, inclusive. Example 1: Input: left = 5, right = 7 Output: 4

Example 2: Input: left = 0, right = 0 Output: 0

Example 3: Input: left = 1, right = 2147483647 Output: 0

Constraints:

0 <= left <= right <= 231 - 1

==
Problem Number: 647 URL: https://leetcode.com/problems/count-primes Title: 204. Count Primes Problem Description: Given an integer n, return the number of prime numbers that are strictly less than n. Example 1: Input: n = 10 Output: 4 Explanation: There are 4 prime numbers less than 10, they are 2, 3, 5, 7.

Example 2: Input: n = 0 Output: 0

Example 3: Input: n = 1 Output: 0

Constraints:

0 <= n <= 5 * 106

==
Problem Number: 648 URL: https://leetcode.com/problems/course-schedule Title: 207. Course Schedule Problem Description: There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.

Return true if you can finish all courses. Otherwise, return false. Example 1: Input: numCourses = 2, prerequisites = [[1,0]] Output: true Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0. So it is possible.

Example 2: Input: numCourses = 2, prerequisites = [[1,0],[0,1]] Output: false Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0, and to take course 0 you should also have finished course 1. So it is impossible.

Constraints:

1 <= numCourses <= 2000 0 <= prerequisites.length <= 5000 prerequisites[i].length == 2 0 <= ai, bi < numCourses All the pairs prerequisites[i] are unique.

==
Problem Number: 649 URL: https://leetcode.com/problems/implement-trie-prefix-tree Title: 208. Implement Trie (Prefix Tree) Problem Description: A trie (pronounced as "try") or prefix tree is a tree data structure used to efficiently store and retrieve keys in a dataset of strings. There are various applications of this data structure, such as autocomplete and spellchecker. Implement the Trie class:

Trie() Initializes the trie object. void insert(String word) Inserts the string word into the trie. boolean search(String word) Returns true if the string word is in the trie (i.e., was inserted before), and false otherwise. boolean startsWith(String prefix) Returns true if there is a previously inserted string word that has the prefix prefix, and false otherwise.

Example 1: Input ["Trie", "insert", "search", "search", "startsWith", "insert", "search"] [[], ["apple"], ["apple"], ["app"], ["app"], ["app"], ["app"]] Output [null, null, true, false, true, null, true]

Explanation Trie trie = new Trie(); trie.insert("apple"); trie.search("apple"); // return True trie.search("app"); // return False trie.startsWith("app"); // return True trie.insert("app"); trie.search("app"); // return True

Constraints:

1 <= word.length, prefix.length <= 2000 word and prefix consist only of lowercase English letters. At most 3 * 104 calls in total will be made to insert, search, and startsWith.

==
Problem Number: 650 URL: https://leetcode.com/problems/minimum-size-subarray-sum Title: 209. Minimum Size Subarray Sum Problem Description: Given an array of positive integers nums and a positive integer target, return the minimal length of a subarray whose sum is greater than or equal to target. If there is no such subarray, return 0 instead. Example 1: Input: target = 7, nums = [2,3,1,2,4,3] Output: 2 Explanation: The subarray [4,3] has the minimal length under the problem constraint.

Example 2: Input: target = 4, nums = [1,4,4] Output: 1

Example 3: Input: target = 11, nums = [1,1,1,1,1,1,1,1] Output: 0

Constraints:

1 <= target <= 109 1 <= nums.length <= 105 1 <= nums[i] <= 104

Follow up: If you have figured out the O(n) solution, try coding another solution of which the time complexity is O(n log(n)). == Problem Number: 651 URL: https://leetcode.com/problems/course-schedule-ii Title: 210. Course Schedule II Problem Description: There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course bi first if you want to take course ai.

For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.

Return the ordering of courses you should take to finish all courses. If there are many valid answers, return any of them. If it is impossible to finish all courses, return an empty array. Example 1: Input: numCourses = 2, prerequisites = [[1,0]] Output: [0,1] Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0. So the correct course order is [0,1].

Example 2: Input: numCourses = 4, prerequisites = [[1,0],[2,0],[3,1],[3,2]] Output: [0,2,1,3] Explanation: There are a total of 4 courses to take. To take course 3 you should have finished both courses 1 and 2. Both courses 1 and 2 should be taken after you finished course 0. So one correct course order is [0,1,2,3]. Another correct ordering is [0,2,1,3].

Example 3: Input: numCourses = 1, prerequisites = [] Output: [0]

Constraints:

1 <= numCourses <= 2000 0 <= prerequisites.length <= numCourses * (numCourses - 1) prerequisites[i].length == 2 0 <= ai, bi < numCourses ai != bi All the pairs [ai, bi] are distinct.

==
Problem Number: 652 URL: https://leetcode.com/problems/design-add-and-search-words-data-structure Title: 211. Design Add and Search Words Data Structure Problem Description: Design a data structure that supports adding new words and finding if a string matches any previously added string. Implement the WordDictionary class:

WordDictionary() Initializes the object. void addWord(word) Adds word to the data structure, it can be matched later. bool search(word) Returns true if there is any string in the data structure that matches word or false otherwise. word may contain dots '.' where dots can be matched with any letter.

Example: Input ["WordDictionary","addWord","addWord","addWord","search","search","search","search"] [[],["bad"],["dad"],["mad"],["pad"],["bad"],[".ad"],["b.."]] Output [null,null,null,null,false,true,true,true]

Explanation WordDictionary wordDictionary = new WordDictionary(); wordDictionary.addWord("bad"); wordDictionary.addWord("dad"); wordDictionary.addWord("mad"); wordDictionary.search("pad"); // return False wordDictionary.search("bad"); // return True wordDictionary.search(".ad"); // return True wordDictionary.search("b.."); // return True

Constraints:

1 <= word.length <= 25 word in addWord consists of lowercase English letters. word in search consist of '.' or lowercase English letters. There will be at most 2 dots in word for search queries. At most 104 calls will be made to addWord and search.

==
Problem Number: 653 URL: https://leetcode.com/problems/house-robber-ii Title: 213. House Robber II Problem Description: You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed. All houses at this place are arranged in a circle. That means the first house is the neighbor of the last one. Meanwhile, adjacent houses have a security system connected, and it will automatically contact the police if two adjacent houses were broken into on the same night. Given an integer array nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police. Example 1: Input: nums = [2,3,2] Output: 3 Explanation: You cannot rob house 1 (money = 2) and then rob house 3 (money = 2), because they are adjacent houses.

Example 2: Input: nums = [1,2,3,1] Output: 4 Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3). Total amount you can rob = 1 + 3 = 4.

Example 3: Input: nums = [1,2,3] Output: 3

Constraints:

1 <= nums.length <= 100 0 <= nums[i] <= 1000

==
Problem Number: 654 URL: https://leetcode.com/problems/kth-largest-element-in-an-array Title: 215. Kth Largest Element in an Array Problem Description: Given an integer array nums and an integer k, return the kth largest element in the array. Note that it is the kth largest element in the sorted order, not the kth distinct element. Can you solve it without sorting? Example 1: Input: nums = [3,2,1,5,6,4], k = 2 Output: 5 Example 2: Input: nums = [3,2,3,1,2,4,5,5,6], k = 4 Output: 4

Constraints:

1 <= k <= nums.length <= 105 -104 <= nums[i] <= 104

==
Problem Number: 655 URL: https://leetcode.com/problems/combination-sum-iii Title: 216. Combination Sum III Problem Description: Find all valid combinations of k numbers that sum up to n such that the following conditions are true:

Only numbers 1 through 9 are used. Each number is used at most once.

Return a list of all possible valid combinations. The list must not contain the same combination twice, and the combinations may be returned in any order. Example 1: Input: k = 3, n = 7 Output: [[1,2,4]] Explanation: 1 + 2 + 4 = 7 There are no other valid combinations. Example 2: Input: k = 3, n = 9 Output: [[1,2,6],[1,3,5],[2,3,4]] Explanation: 1 + 2 + 6 = 9 1 + 3 + 5 = 9 2 + 3 + 4 = 9 There are no other valid combinations.

Example 3: Input: k = 4, n = 1 Output: [] Explanation: There are no valid combinations. Using 4 different numbers in the range [1,9], the smallest sum we can get is 1+2+3+4 = 10 and since 10 > 1, there are no valid combination.

Constraints:

2 <= k <= 9 1 <= n <= 60

==
Problem Number: 656 URL: https://leetcode.com/problems/maximal-square Title: 221. Maximal Square Problem Description: Given an m x n binary matrix filled with 0's and 1's, find the largest square containing only 1's and return its area. Example 1:

Input: matrix = [["1","0","1","0","0"],["1","0","1","1","1"],["1","1","1","1","1"],["1","0","0","1","0"]] Output: 4

Example 2:

Input: matrix = [["0","1"],["1","0"]] Output: 1

Example 3: Input: matrix = [["0"]] Output: 0

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 300 matrix[i][j] is '0' or '1'.

==
Problem Number: 657 URL: https://leetcode.com/problems/rectangle-area Title: 223. Rectangle Area Problem Description: Given the coordinates of two rectilinear rectangles in a 2D plane, return the total area covered by the two rectangles. The first rectangle is defined by its bottom-left corner (ax1, ay1) and its top-right corner (ax2, ay2). The second rectangle is defined by its bottom-left corner (bx1, by1) and its top-right corner (bx2, by2). Example 1:

Input: ax1 = -3, ay1 = 0, ax2 = 3, ay2 = 4, bx1 = 0, by1 = -1, bx2 = 9, by2 = 2 Output: 45

Example 2: Input: ax1 = -2, ay1 = -2, ax2 = 2, ay2 = 2, bx1 = -2, by1 = -2, bx2 = 2, by2 = 2 Output: 16

Constraints:

-104 <= ax1 <= ax2 <= 104 -104 <= ay1 <= ay2 <= 104 -104 <= bx1 <= bx2 <= 104 -104 <= by1 <= by2 <= 104

==
Problem Number: 658 URL: https://leetcode.com/problems/basic-calculator-ii Title: 227. Basic Calculator II Problem Description: Given a string s which represents an expression, evaluate this expression and return its value. The integer division should truncate toward zero. You may assume that the given expression is always valid. All intermediate results will be in the range of [-231, 231 - 1]. Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval(). Example 1: Input: s = "3+2*2" Output: 7 Example 2: Input: s = " 3/2 " Output: 1 Example 3: Input: s = " 3+5 / 2 " Output: 5

Constraints:

1 <= s.length <= 3 * 105 s consists of integers and operators ('+', '-', '*', '/') separated by some number of spaces. s represents a valid expression. All the integers in the expression are non-negative integers in the range [0, 231 - 1]. The answer is guaranteed to fit in a 32-bit integer.

==
Problem Number: 659 URL: https://leetcode.com/problems/majority-element-ii Title: 229. Majority Element II Problem Description: Given an integer array of size n, find all elements that appear more than n/3 times. Example 1: Input: nums = [3,2,3] Output: [3]

Example 2: Input: nums = [1] Output: [1]

Example 3: Input: nums = [1,2] Output: [1,2]

Constraints:

1 <= nums.length <= 5 * 104 -109 <= nums[i] <= 109

Follow up: Could you solve the problem in linear time and in O(1) space?

==
Problem Number: 660 URL: https://leetcode.com/problems/kth-smallest-element-in-a-bst Title: 230. Kth Smallest Element in a BST Problem Description: Given the root of a binary search tree, and an integer k, return the kth smallest value (1-indexed) of all the values of the nodes in the tree. Example 1:

Input: root = [3,1,4,null,2], k = 1 Output: 1

Example 2:

Input: root = [5,3,6,2,4,null,null,1], k = 3 Output: 3

Constraints:

The number of nodes in the tree is n. 1 <= k <= n <= 104 0 <= Node.val <= 104

Follow up: If the BST is modified often (i.e., we can do insert and delete operations) and you need to find the kth smallest frequently, how would you optimize?

==
Problem Number: 661 URL: https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree Title: 235. Lowest Common Ancestor of a Binary Search Tree Problem Description: Given a binary search tree (BST), find the lowest common ancestor (LCA) node of two given nodes in the BST. According to the definition of LCA on Wikipedia: "The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow a node to be a descendant of itself)." Example 1:

Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8 Output: 6 Explanation: The LCA of nodes 2 and 8 is 6.

Example 2:

Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4 Output: 2 Explanation: The LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

Example 3: Input: root = [2,1], p = 2, q = 1 Output: 2

Constraints:

The number of nodes in the tree is in the range [2, 105]. -109 <= Node.val <= 109 All Node.val are unique. p != q p and q will exist in the BST.

==
Problem Number: 662 URL: https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree Title: 236. Lowest Common Ancestor of a Binary Tree Problem Description: Given a binary tree, find the lowest common ancestor (LCA) of two given nodes in the tree. According to the definition of LCA on Wikipedia: "The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow a node to be a descendant of itself)." Example 1:

Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1 Output: 3 Explanation: The LCA of nodes 5 and 1 is 3.

Example 2:

Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 4 Output: 5 Explanation: The LCA of nodes 5 and 4 is 5, since a node can be a descendant of itself according to the LCA definition.

Example 3: Input: root = [1,2], p = 1, q = 2 Output: 1

Constraints:

The number of nodes in the tree is in the range [2, 105]. -109 <= Node.val <= 109 All Node.val are unique. p != q p and q will exist in the tree.

==

Problem Number: 663 URL: https://leetcode.com/problems/delete-node-in-a-linked-list Title: 237. Delete Node in a Linked List Problem Description: There is a singly-linked list head and we want to delete a node node in it. You are given the node to be deleted node. You will not be given access to the first node of head. All the values of the linked list are unique, and it is guaranteed that the given node node is not the last node in the linked list. Delete the given node. Note that by deleting the node, we do not mean removing it from memory. We mean:

The value of the given node should not exist in the linked list. The number of nodes in the linked list should decrease by one. All the values before node should be in the same order. All the values after node should be in the same order.

Custom testing:

For the input, you should provide the entire linked list head and the node to be given node. node should not be the last node of the list and should be an actual node in the list. We will build the linked list and pass the node to your function. The output will be the entire list after calling your function.

Example 1:

Input: head = [4,5,1,9], node = 5 Output: [4,1,9] Explanation: You are given the second node with value 5, the linked list should become 4 -> 1 -> 9 after calling your function.

Example 2:

Input: head = [4,5,1,9], node = 1 Output: [4,5,9] Explanation: You are given the third node with value 1, the linked list should become 4 -> 5 -> 9 after calling your function.

Constraints:

The number of the nodes in the given list is in the range [2, 1000]. -1000 <= Node.val <= 1000 The value of each node in the list is unique. The node to be deleted is in the list and is not a tail node.

==

Problem Number: 664 URL: https://leetcode.com/problems/product-of-array-except-self Title: 238. Product of Array Except Self Problem Description: Given an integer array nums, return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i]. The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer. You must write an algorithm that runs in O(n) time and without using the division operation. Example 1: Input: nums = [1,2,3,4] Output: [24,12,8,6] Example 2: Input: nums = [-1,1,0,-3,3] Output: [0,0,9,0,0]

Constraints:

2 <= nums.length <= 105 -30 <= nums[i] <= 30 The product of any prefix or suffix of nums is guaranteed to fit in a 32-bit integer.

Follow up: Can you solve the problem in O(1) extra space complexity? (The output array does not count as extra space for space complexity analysis.)

==
Problem Number: 665 URL: https://leetcode.com/problems/search-a-2d-matrix-ii Title: 240. Search a 2D Matrix II Problem Description: Write an efficient algorithm that searches for a value target in an m x n integer matrix matrix. This matrix has the following properties:

Integers in each row are sorted in ascending from left to right. Integers in each column are sorted in ascending from top to bottom.

Example 1:

Input: matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]], target = 5 Output: true

Example 2:

Input: matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]], target = 20 Output: false

Constraints:

m == matrix.length n == matrix[i].length 1 <= n, m <= 300 -109 <= matrix[i][j] <= 109 All the integers in each row are sorted in ascending order. All the integers in each column are sorted in ascending order. -109 <= target <= 109

==
Problem Number: 666 URL: https://leetcode.com/problems/different-ways-to-add-parentheses Title: 241. Different Ways to Add Parentheses Problem Description: Given a string expression of numbers and operators, return all possible results from computing all the different possible ways to group numbers and operators. You may return the answer in any order. The test cases are generated such that the output values fit in a 32-bit integer and the number of different results does not exceed 104. Example 1: Input: expression = "2-1-1" Output: [0,2] Explanation: ((2-1)-1) = 0 (2-(1-1)) = 2

Example 2: Input: expression = "2*3-4*5" Output: [-34,-14,-10,-10,10] Explanation: (2*(3-(4*5))) = -34 ((2*3)-(4*5)) = -14 ((2*(3-4))*5) = -10 (2*((3-4)*5)) = -10 (((2*3)-4)*5) = 10

Constraints:

1 <= expression.length <= 20 expression consists of digits and the operator '+', '-', and '*'. All the integer values in the input expression are in the range [0, 99].

==

Problem Number: 667 URL: https://leetcode.com/problems/single-number-iii Title: 260. Single Number III Problem Description: Given an integer array nums, in which exactly two elements appear only once and all the other elements appear exactly twice. Find the two elements that appear only once. You can return the answer in any order. You must write an algorithm that runs in linear runtime complexity and uses only constant extra space. Example 1: Input: nums = [1,2,1,3,2,5] Output: [3,5] Explanation: [5, 3] is also a valid answer.

Example 2: Input: nums = [-1,0] Output: [-1,0]

Example 3: Input: nums = [0,1] Output: [1,0]

Constraints:

2 <= nums.length <= 3 * 104 -231 <= nums[i] <= 231 - 1 Each integer in nums will appear twice, only two integers will appear once.

==

Problem Number: 668 URL: https://leetcode.com/problems/ugly-number-ii Title: 264. Ugly Number II Problem Description: An ugly number is a positive integer whose prime factors are limited to 2, 3, and 5. Given an integer n, return the nth ugly number. Example 1: Input: n = 10 Output: 12 Explanation: [1, 2, 3, 4, 5, 6, 8, 9, 10, 12] is the sequence of the first 10 ugly numbers.

Example 2: Input: n = 1 Output: 1 Explanation: 1 has no prime factors, therefore all of its prime factors are limited to 2, 3, and 5.

Constraints:

1 <= n <= 1690

==

Problem Number: 669 URL: https://leetcode.com/problems/h-index Title: 274. H-Index Problem Description: Given an array of integers citations where citations[i] is the number of citations a researcher received for their ith paper, return the researcher's h-index. According to the definition of h-index on Wikipedia: The h-index is defined as the maximum value of h such that the given researcher has published at least h papers that have each been cited at least h times. Example 1: Input: citations = [3,0,6,1,5] Output: 3 Explanation: [3,0,6,1,5] means the researcher has 5 papers in total and each of them had received 3, 0, 6, 1, 5 citations respectively. Since the researcher has 3 papers with at least 3 citations each and the remaining two with no more than 3 citations each, their h-index is 3.

Example 2: Input: citations = [1,3,1] Output: 1

Constraints:

n == citations.length 1 <= n <= 5000 0 <= citations[i] <= 1000

==

Problem Number: 670 URL: https://leetcode.com/problems/h-index-ii Title: 275. H-Index II Problem Description: Given an array of integers citations where citations[i] is the number of citations a researcher received for their ith paper and citations is sorted in ascending order, return the researcher's h-index. According to the definition of h-index on Wikipedia: The h-index is defined as the maximum value of h such that the given researcher has published at least h papers that have each been cited at least h times. You must write an algorithm that runs in logarithmic time.   Example 1: Input: citations = [0,1,3,5,6] Output: 3 Explanation: [0,1,3,5,6] means the researcher has 5 papers in total and each of them had received 0, 1, 3, 5, 6 citations respectively. Since the researcher has 3 papers with at least 3 citations each and the remaining two with no more than 3 citations each, their h-index is 3.

Example 2: Input: citations = [1,2,100] Output: 2

Constraints:

n == citations.length 1 <= n <= 105 0 <= citations[i] <= 1000 citations is sorted in ascending order.

==

Problem Number: 671 URL: https://leetcode.com/problems/perfect-squares Title: 279. Perfect Squares Problem Description: Given an integer n, return the least number of perfect square numbers that sum to n. A perfect square is an integer that is the square of an integer; in other words, it is the product of some integer with itself. For example, 1, 4, 9, and 16 are perfect squares while 3 and 11 are not.   Example 1: Input: n = 12 Output: 3 Explanation: 12 = 4 + 4 + 4.

Example 2: Input: n = 13 Output: 2 Explanation: 13 = 4 + 9.

Constraints:

1 <= n <= 104

==

Problem Number: 672 URL: https://leetcode.com/problems/peeking-iterator Title: 284. Peeking Iterator Problem Description: Design an iterator that supports the peek operation on an existing iterator in addition to the hasNext and the next operations. Implement the PeekingIterator class:

PeekingIterator(Iterator<int> nums) Initializes the object with the given integer iterator iterator. int next() Returns the next element in the array and moves the pointer to the next element. boolean hasNext() Returns true if there are still elements in the array. int peek() Returns the next element in the array without moving the pointer.

Note: Each language may have a different implementation of the constructor and Iterator, but they all support the int next() and boolean hasNext() functions. Example 1: Input ["PeekingIterator", "next", "peek", "next", "next", "hasNext"] [[[1, 2, 3]], [], [], [], [], []] Output [null, 1, 2, 2, 3, false]

Explanation PeekingIterator peekingIterator = new PeekingIterator([1, 2, 3]); // [1,2,3] peekingIterator.next(); // return 1, the pointer moves to the next element [1,2,3]. peekingIterator.peek(); // return 2, the pointer does not move [1,2,3]. peekingIterator.next(); // return 2, the pointer moves to the next element [1,2,3] peekingIterator.next(); // return 3, the pointer moves to the next element [1,2,3] peekingIterator.hasNext(); // return False

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 1000 All the calls to next and peek are valid. At most 1000 calls will be made to next, hasNext, and peek.

Follow up: How would you extend your design to be generic and work with all types, not just integer? ==
Problem Number: 673 URL: https://leetcode.com/problems/find-the-duplicate-number Title: 287. Find the Duplicate Number Problem Description: Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive. There is only one repeated number in nums, return this repeated number. You must solve the problem without modifying the array nums and uses only constant extra space. Example 1: Input: nums = [1,3,4,2,2] Output: 2

Example 2: Input: nums = [3,1,3,4,2] Output: 3

Constraints:

1 <= n <= 105 nums.length == n + 1 1 <= nums[i] <= n All the integers in nums appear only once except for precisely one integer which appears two or more times.

Follow up:

How can we prove that at least one duplicate number must exist in nums? Can you solve the problem in linear runtime complexity?

==
Problem Number: 674 URL: https://leetcode.com/problems/game-of-life Title: 289. Game of Life Problem Description: According to Wikipedia's article: "The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970." The board is made up of an m x n grid of cells, where each cell has an initial state: live (represented by a 1) or dead (represented by a 0). Each cell interacts with its

eight neighbors (horizontal, vertical, diagonal) using the following four rules (taken from the above Wikipedia article):

Any live cell with fewer than two live neighbors dies as if caused by under-population. Any live cell with two or three live neighbors lives on to the next generation. Any live cell with more than three live neighbors dies, as if by over-population. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The next state is created by applying the above rules simultaneously to every cell in the current state, where births and deaths occur simultaneously. Given the current state of the m x n grid board, return the next state. Example 1:

Input: board = [[0,1,0],[0,0,1],[1,1,1],[0,0,0]] Output: [[0,0,0],[1,0,1],[0,1,1],[0,1,0]]

Example 2:

Input: board = [[1,1],[1,0]] Output: [[1,1],[1,1]]

Constraints:

m == board.length n == board[i].length 1 <= m, n <= 25 board[i][j] is 0 or 1.

Follow up:

Could you solve it in-place? Remember that the board needs to be updated simultaneously: You cannot update some cells first and then use their updated values to update other cells. In this question, we represent the board using a 2D array. In principle, the board is infinite, which would cause problems when the active area encroaches upon the border of the array (i.e., live cells reach the border). How would you address these problems?

==
Problem Number: 675 URL: https://leetcode.com/problems/bulls-and-cows Title: 299. Bulls and Cows Problem Description: You are playing the Bulls and Cows game with your friend. You write down a secret number and ask your friend to guess what the number is. When your friend makes a guess, you provide a hint with the following info:

The number of "bulls", which are digits in the guess that are in the correct position. The number of "cows", which are digits in the guess that are in your secret number but are located in the wrong position. Specifically, the non-bull digits in the guess that could be rearranged such that they become bulls.

Given the secret number secret and your friend's guess guess, return the hint for your friend's guess. The hint should be formatted as "xAyB", where x is the number of bulls and y is the number of cows. Note that both secret and guess may contain duplicate digits. Example 1: Input: secret = "1807", guess = "7810" Output: "1A3B" Explanation: Bulls are connected with a '|' and cows are underlined: "1807" | "7810" Example 2: Input: secret = "1123", guess = "0111" Output: "1A1B" Explanation: Bulls are connected with a '|' and cows are underlined: "1123" "1123" | or | "0111" "0111" Note that only one of the two unmatched 1s is counted as a cow since the non-bull digits can only be rearranged to allow one 1 to be a bull.

Constraints:

1 <= secret.length, guess.length <= 1000 secret.length == guess.length secret and guess consist of digits only.

==
Problem Number: 676 URL: https://leetcode.com/problems/longest-increasing-subsequence Title: 300. Longest Increasing Subsequence Problem Description: Given an integer array nums, return the length of the longest strictly increasing subsequence. Example 1: Input: nums = [10,9,2,5,3,7,101,18] Output: 4 Explanation: The longest increasing subsequence is [2,3,7,101], therefore the length is 4.

Example 2: Input: nums = [0,1,0,3,2,3] Output: 4

Example 3: Input: nums = [7,7,7,7,7,7,7] Output: 1

Constraints:

1 <= nums.length <= 2500 -104 <= nums[i] <= 104

Follow up: Can you come up with an algorithm that runs in O(n log(n)) time complexity?

==
Problem Number: 677 URL: https://leetcode.com/problems/range-sum-query-2d-immutable Title: 304. Range Sum Query 2D - Immutable Problem Description: Given a 2D matrix matrix, handle multiple queries of the following type:

Calculate the sum of the elements of matrix inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

Implement the NumMatrix class:

NumMatrix(int[][] matrix) Initializes the object with the integer matrix matrix. int sumRegion(int row1, int col1, int row2, int col2) Returns the sum of the elements of matrix inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

You must design an algorithm where sumRegion works on O(1) time complexity. Example 1:

Input ["NumMatrix", "sumRegion", "sumRegion", "sumRegion"] [[[[3, 0, 1, 4, 2], [5, 6, 3, 2, 1], [1, 2, 0, 1, 5], [4, 1, 0, 1, 7], [1, 0, 3, 0, 5]]], [2, 1, 4, 3], [1, 1, 2, 2], [1, 2, 2, 4]] Output [null, 8, 11, 12]

Explanation NumMatrix numMatrix = new NumMatrix([[3, 0, 1, 4, 2], [5, 6, 3, 2, 1], [1, 2, 0, 1, 5], [4, 1, 0, 1, 7], [1, 0, 3, 0, 5]]); numMatrix.sumRegion(2, 1, 4, 3); // return 8 (i.e sum of the red rectangle) numMatrix.sumRegion(1, 1, 2, 2);

// return 11 (i.e sum of the green rectangle) numMatrix.sumRegion(1, 2, 2, 4); // return 12 (i.e sum of the blue rectangle)

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 200 -104 <= matrix[i][j] <= 104 0 <= row1 <= row2 < m 0 <= col1 <= col2 < n At most 104 calls will be made to sumRegion.

==

Problem Number: 678 URL: https://leetcode.com/problems/additive-number Title: 306. Additive Number Problem Description: An additive number is a string whose digits can form an additive sequence. A valid additive sequence should contain at least three numbers. Except for the first two numbers, each subsequent number in the sequence must be the sum of the preceding two. Given a string containing only digits, return true if it is an additive number or false otherwise. Note: Numbers in the additive sequence cannot have leading zeros, so sequence 1, 2, 03 or 1, 02, 3 is invalid. Example 1: Input: "112358" Output: true Explanation: The digits can form an additive sequence: 1, 1, 2, 3, 5, 8. 1 + 1 = 2, 1 + 2 = 3, 2 + 3 = 5, 3 + 5 = 8

Example 2: Input: "199100199" Output: true Explanation: The additive sequence is: 1, 99, 100, 199. 1 + 99 = 100, 99 + 100 = 199

Constraints:

1 <= num.length <= 35 num consists only of digits.

Follow up: How would you handle overflow for very large input integers?

==

Problem Number: 679 URL: https://leetcode.com/problems/range-sum-query-mutable Title: 307. Range Sum Query - Mutable Problem Description: Given an integer array nums, handle multiple queries of the following types:

Update the value of an element in nums. Calculate the sum of the elements of nums between indices left and right inclusive where left <= right.

Implement the NumArray class:

NumArray(int[] nums) Initializes the object with the integer array nums. void update(int index, int val) Updates the value of nums[index] to be val. int sumRange(int left, int right) Returns the sum of the elements of nums between indices left and right inclusive (i.e. nums[left] + nums[left + 1] + ... + nums[right]).

Example 1: Input ["NumArray", "sumRange", "update", "sumRange"] [[[1, 3, 5]], [0, 2], [1, 2], [0, 2]] Output [null, 9, null, 8]

Explanation NumArray numArray = new NumArray([1, 3, 5]); numArray.sumRange(0, 2); // return 1 + 3 + 5 = 9 numArray.update(1, 2); // nums = [1, 2, 5] numArray.sumRange(0, 2); // return 1 + 2 + 5 = 8

Constraints:

1 <= nums.length <= 3 * 104 -100 <= nums[i] <= 100 0 <= index < nums.length -100 <= val <= 100 0 <= left <= right < nums.length At most 3 * 104 calls will be made to update and sumRange.

==
Problem Number: 680 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-cooldown Title: 309. Best Time to Buy and Sell Stock with Cooldown Problem Description: You are given an array prices where prices[i] is the price of a given stock on the ith day. Find the maximum profit you can achieve. You may complete as many transactions as you like (i.e., buy one and sell one share of the stock multiple times) with the following restrictions:

After you sell your stock, you cannot buy stock on the next day (i.e., cooldown one day).

Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again). Example 1: Input: prices = [1,2,3,0,2] Output: 3 Explanation: transactions = [buy, sell, cooldown, buy, sell]

Example 2: Input: prices = [1] Output: 0

Constraints:

1 <= prices.length <= 5000 0 <= prices[i] <= 1000

==
Problem Number: 681 URL: https://leetcode.com/problems/minimum-height-trees Title: 310. Minimum Height Trees Problem Description: A tree is an undirected graph in which any two vertices are connected by exactly one path. In other words, any connected graph without simple cycles is a tree. Given a tree of n nodes labelled from 0 to n - 1, and an array of n - 1 edges where edges[i] = [ai, bi] indicates that there is an undirected edge between the two nodes ai and bi in the tree, you can choose any node of the tree as the root. When you select a node x as the root, the result tree has height h. Among all possible rooted trees, those with minimum height (i.e. min(h)) are called minimum height trees (MHTs). Return a list of all MHTs' root labels. You can return the answer in any order. The height of a rooted tree is the number of edges on the longest downward path between the root and a leaf. Example 1:

Input: n = 4, edges = [[1,0],[1,2],[1,3]] Output: [1] Explanation: As shown, the height of the tree is 1 when the root is the node with label 1 which is the only MHT.

Example 2:

Input: n = 6, edges = [[3,0],[3,1],[3,2],[3,4],[5,4]] Output: [3,4]

Constraints:

1 <= n <= 2 * 104 edges.length == n - 1 0 <= ai, bi < n ai != bi All the pairs (ai, bi) are distinct. The given input is guaranteed to be a tree and there will be no repeated edges.

==

Problem Number: 682 URL: https://leetcode.com/problems/super-ugly-number Title: 313. Super Ugly Number Problem Description: A super ugly number is a positive integer whose prime factors are in the array primes. Given an integer n and an array of integers primes, return the nth super ugly number. The nth super ugly number is guaranteed to fit in a 32-bit signed integer. Example 1: Input: n = 12, primes = [2,7,13,19] Output: 32 Explanation: [1,2,4,7,8,13,14,16,19,26,28,32] is the sequence of the first 12 super ugly numbers given primes = [2,7,13,19].

Example 2: Input: n = 1, primes = [2,3,5] Output: 1 Explanation: 1 has no prime factors, therefore all of its prime factors are in the array primes = [2,3,5].

Constraints:

1 <= n <= 105 1 <= primes.length <= 100 2 <= primes[i] <= 1000 primes[i] is guaranteed to be a prime number. All the values of primes are unique and sorted in ascending order.

==

Problem Number: 683 URL: https://leetcode.com/problems/remove-duplicate-letters Title: 316. Remove Duplicate Letters Problem Description: Given a string s, remove duplicate letters so that every letter appears once and only once. You must make sure your result is the smallest in lexicographical order among all possible results. Example 1: Input: s = "bcabc" Output: "abc"

Example 2: Input: s = "cbacdcbc" Output: "acdb"

Constraints:

1 <= s.length <= 104 s consists of lowercase English letters.

Note: This question is the same as 1081: https://leetcode.com/problems/smallest-subsequence-of-distinct-characters/

==

Problem Number: 684 URL: https://leetcode.com/problems/maximum-product-of-word-lengths Title: 318. Maximum Product of Word Lengths Problem Description: Given a string array words, return the maximum value of length(word[i]) * length(word[j]) where the two words do not share common letters. If no such two words exist, return 0. Example 1: Input: words = ["abcw","baz","foo","bar","xtfn","abcdef"] Output: 16 Explanation: The two words can be "abcw", "xtfn".

Example 2: Input: words = ["a","ab","abc","d","cd","bcd","abcd"] Output: 4 Explanation: The two words can be "ab", "cd".

Example 3: Input: words = ["a","aa","aaa","aaaa"] Output: 0 Explanation: No such pair of words.

Constraints:

2 <= words.length <= 1000 1 <= words[i].length <= 1000 words[i] consists only of lowercase English letters.

==
Problem Number: 685 URL: https://leetcode.com/problems/bulb-switcher Title: 319. Bulb Switcher Problem Description: There are n bulbs that are initially off. You first turn on all the bulbs, then you turn off every second bulb. On the third round, you toggle every third bulb (turning on if it's off or turning off if it's on). For the ith round, you toggle every i bulb. For the nth round, you only toggle the last bulb. Return the number of bulbs that are on after n rounds. Example 1:

Input: n = 3 Output: 1 Explanation: At first, the three bulbs are [off, off, off]. After the first round, the three bulbs are [on, on, on]. After the second round, the three bulbs are [on, off, on]. After the third round, the three bulbs are [on, off, off]. So you should return 1 because there is only one bulb is on. Example 2: Input: n = 0 Output: 0

Example 3: Input: n = 1 Output: 1

Constraints:

0 <= n <= 109

==
Problem Number: 686 URL: https://leetcode.com/problems/coin-change Title: 322. Coin Change Problem Description: You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money. Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1. You may assume that you have an infinite number of each kind of coin. Example 1: Input: coins = [1,2,5], amount = 11 Output: 3 Explanation: 11 = 5 + 5 + 1

Example 2: Input: coins = [2], amount = 3 Output: -1

Example 3: Input: coins = [1], amount = 0 Output: 0

Constraints:

1 <= coins.length <= 12 1 <= coins[i] <= 231 - 1 0 <= amount <= 104

==
Problem Number: 687 URL: https://leetcode.com/problems/wiggle-sort-ii Title: 324. Wiggle Sort II Problem Description: Given an integer array nums, reorder it such that nums[0] < nums[1] > nums[2] < nums[3].... You may

assume the input array always has a valid answer.   Example 1: Input: nums = [1,5,1,1,6,4] Output: [1,6,1,5,1,4] Explanation: [1,4,1,5,1,6] is also accepted.

Example 2: Input: nums = [1,3,2,2,3,1] Output: [2,3,1,3,1,2]

Constraints:

1 <= nums.length <= 5 * 104 0 <= nums[i] <= 5000 It is guaranteed that there will be an answer for the given input nums.

Follow Up: Can you do it in O(n) time and/or in-place with O(1) extra space?

==

Problem Number: 688 URL: https://leetcode.com/problems/odd-even-linked-list Title: 328. Odd Even Linked List Problem Description: Given the head of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return the reordered list. The first node is considered odd, and the second node is even, and so on. Note that the relative order inside both the even and odd groups should remain as it was in the input. You must solve the problem in O(1) extra space complexity and O(n) time complexity.   Example 1:

Input: head = [1,2,3,4,5] Output: [1,3,5,2,4]

Example 2:

Input: head = [2,1,3,5,6,4,7] Output: [2,3,6,7,1,5,4]

Constraints:

The number of nodes in the linked list is in the range [0, 104]. -106 <= Node.val <= 106

==

Problem Number: 689 URL: https://leetcode.com/problems/verify-preorder-serialization-of-a-binary-tree Title: 331. Verify Preorder Serialization of a Binary Tree Problem Description: One way to serialize a binary tree is to use preorder traversal. When we encounter a non-null node, we record the node's value. If it is a null node, we record using a sentinel value such as '#'.

For example, the above binary tree can be serialized to the string "9,3,4,#,#,1,#,#,2,#,6,#,#", where '#' represents a null node. Given a string of comma-separated values preorder, return true if it is a correct preorder traversal serialization of a binary tree. It is guaranteed that each comma-separated value in the string must be either an integer or a character '#' representing null pointer. You may assume that the input format is always valid.

For example, it could never contain two consecutive commas, such as "1,,3".

Note: You are not allowed to reconstruct the tree.   Example 1: Input: preorder = "9,3,4,#,#,1,#,#,2,#,6,#,#" Output: true Example 2: Input: preorder = "1,#" Output: false Example 3: Input: preorder = "9,#,#,1" Output: false

Constraints:

1 <= preorder.length <= 104 preorder consist of integers in the range [0, 100] and '#' separated by commas ','.

==
Problem Number: 690 URL: https://leetcode.com/problems/increasing-triplet-subsequence Title: 334. Increasing Triplet Subsequence Problem Description: Given an integer array nums, return true if there exists a triple of indices (i, j, k) such that i < j < k and nums[i] < nums[j] < nums[k]. If no such indices exists, return false.   Example 1: Input: nums = [1,2,3,4,5] Output: true Explanation: Any triplet where i < j < k is valid.

Example 2: Input: nums = [5,4,3,2,1] Output: false Explanation: No triplet exists.

Example 3: Input: nums = [2,1,5,0,4,6] Output: true Explanation: The triplet (3, 4, 5) is valid because nums[3] == 0 < nums[4] == 4 < nums[5] == 6.

Constraints:

1 <= nums.length <= 5 * 105 -231 <= nums[i] <= 231 - 1

Follow up: Could you implement a solution that runs in O(n) time complexity and O(1) space complexity? ==
Problem Number: 691 URL: https://leetcode.com/problems/house-robber-iii Title: 337. House Robber III Problem Description: The thief has found himself a new place for his thievery again. There is only one entrance to this area, called root. Besides the root, each house has one and only one parent house. After a tour, the smart thief realized that all houses in this place form a binary tree. It will automatically contact the police if two directly-linked houses were broken into on the same night. Given the root of the binary tree, return the maximum amount of money the thief can rob without alerting the police. Example 1:

Input: root = [3,2,3,null,3,null,1] Output: 7 Explanation: Maximum amount of money the thief can rob = 3 + 3 + 1 = 7.

Example 2:

Input: root = [3,4,5,1,3,null,1] Output: 9 Explanation: Maximum amount of money the thief can rob = 4 + 5 = 9.

Constraints:

The number of nodes in the tree is in the range [1, 104]. 0 <= Node.val <= 104

==
Problem Number: 692 URL: https://leetcode.com/problems/flatten-nested-list-iterator Title: 341. Flatten Nested List Iterator Problem Description: You are given a nested list of integers nestedList. Each element is either an integer

or a list whose elements may also be integers or other lists. Implement an iterator to flatten it. Implement the NestedIterator class:

NestedIterator(List<NestedInteger> nestedList) Initializes the iterator with the nested list nestedList. int next() Returns the next integer in the nested list. boolean hasNext() Returns true if there are still some integers in the nested list and false otherwise.

Your code will be tested with the following pseudocode: initialize iterator with nestedList res = [] while iterator.hasNext() append iterator.next() to the end of res return res

If res matches the expected flattened list, then your code will be judged as correct. Example 1: Input: nestedList = [[1,1],2,[1,1]] Output: [1,1,2,1,1] Explanation: By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,1,2,1,1].

Example 2: Input: nestedList = [1,[4,[6]]] Output: [1,4,6] Explanation: By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,4,6].

Constraints:

1 <= nestedList.length <= 500 The values of the integers in the nested list is in the range [-106, 106].

==
Problem Number: 693 URL: https://leetcode.com/problems/integer-break Title: 343. Integer Break Problem Description: Given an integer n, break it into the sum of k positive integers, where k >= 2, and maximize the product of those integers. Return the maximum product you can get. Example 1: Input: n = 2 Output: 1 Explanation: 2 = 1 + 1, 1 × 1 = 1.

Example 2: Input: n = 10 Output: 36 Explanation: 10 = 3 + 3 + 4, 3 × 3 × 4 = 36.

Constraints:

2 <= n <= 58

==
Problem Number: 694 URL: https://leetcode.com/problems/top-k-frequent-elements Title: 347. Top K Frequent Elements Problem Description: Given an integer array nums and an integer k, return the k most frequent elements. You may return the answer in any order. Example 1: Input: nums = [1,1,1,2,2,3], k = 2 Output: [1,2] Example 2: Input: nums = [1], k = 1 Output: [1]

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104 k is in the range [1, the number of unique elements in the array]. It is guaranteed that the answer is unique.

Follow up: Your algorithm's time complexity must be better than O(n log n), where n is the array's size.

== Problem Number: 695 URL: https://leetcode.com/problems/design-twitter Title: 355. Design Twitter Problem Description: Design a simplified version of Twitter where users can post tweets, follow/unfollow another user, and is able to see the 10 most recent tweets in the user's news feed. Implement the Twitter class:

Twitter() Initializes your twitter object. void postTweet(int userId, int tweetId) Composes a new tweet with ID tweetId by the user userId. Each call to this function will be made with a unique tweetId. List<Integer> getNewsFeed(int userId) Retrieves the 10 most recent tweet IDs in the user's news feed. Each item in the news feed must be posted by users who the user followed or by the user themself. Tweets must be ordered from most recent to least recent. void follow(int followerId, int followeeId) The user with ID followerId started following the user with ID followeeId. void unfollow(int followerId, int followeeId) The user with ID followerId started unfollowing the user with ID followeeId.

Example 1: Input ["Twitter", "postTweet", "getNewsFeed", "follow", "postTweet", "getNewsFeed", "unfollow", "getNewsFeed"] [[], [1, 5], [1], [1, 2], [2, 6], [1], [1, 2], [1]] Output [null, null, [5], null, null, [6, 5], null, [5]]

Explanation Twitter twitter = new Twitter(); twitter.postTweet(1, 5); // User 1 posts a new tweet (id = 5). twitter.getNewsFeed(1); // User 1's news feed should return a list with 1 tweet id -> [5]. return [5] twitter.follow(1, 2); // User 1 follows user 2. twitter.postTweet(2, 6); // User 2 posts a new tweet (id = 6). twitter.getNewsFeed(1); // User 1's news feed should return a list with 2 tweet ids -> [6, 5]. Tweet id 6 should precede tweet id 5 because it is posted after tweet id 5. twitter.unfollow(1, 2); // User 1 unfollows user 2. twitter.getNewsFeed(1); // User 1's news feed should return a list with 1 tweet id -> [5], since user 1 is no longer following user 2.

Constraints:

1 <= userId, followerId, followeeId <= 500 0 <= tweetId <= 104 All the tweets have unique IDs. At most 3 * 104 calls will be made to postTweet, getNewsFeed, follow, and unfollow.

== Problem Number: 696 URL: https://leetcode.com/problems/count-numbers-with-unique-digits Title: 357. Count Numbers with Unique Digits Problem Description: Given an integer n, return the count of all numbers with unique digits, x, where 0 <= x < 10n. Example 1: Input: n = 2 Output: 91 Explanation: The answer should be the total numbers in the range of 0 x < 100, excluding 11,22,33,44,55,66,77,88,99

Example 2: Input: n = 0 Output: 1

Constraints:

0 <= n <= 8

== Problem Number: 697 URL: https://leetcode.com/problems/water-and-jug-problem Title: 365. Water and Jug Problem Problem Description: You are given two jugs with capacities jug1Capacity and jug2Capacity liters. There is an infinite amount of water supply available. Determine whether it is possible to measure exactly targetCapacity liters using these two jugs. If targetCapacity liters of water are measurable, you must have targetCapacity liters of water contained within one or both buckets by the end. Operations allowed:

Fill any of the jugs with water. Empty any of the jugs. Pour water from one jug into another till the other jug is completely full, or the first jug itself is empty.

Example 1: Input: jug1Capacity = 3, jug2Capacity = 5, targetCapacity = 4 Output: true Explanation: The famous Die Hard example

Example 2: Input: jug1Capacity = 2, jug2Capacity = 6, targetCapacity = 5 Output: false

Example 3: Input: jug1Capacity = 1, jug2Capacity = 2, targetCapacity = 3 Output: true

Constraints:

1 <= jug1Capacity, jug2Capacity, targetCapacity <= 106

== Problem Number: 698 URL: https://leetcode.com/problems/largest-divisible-subset Title: 368. Largest Divisible Subset Problem Description: Given a set of distinct positive integers nums, return the largest subset answer such that every pair (answer[i], answer[j]) of elements in this subset satisfies:

answer[i] % answer[j] == 0, or answer[j] % answer[i] == 0

If there are multiple solutions, return any of them. Example 1: Input: nums = [1,2,3] Output: [1,2] Explanation: [1,3] is also accepted.

Example 2: Input: nums = [1,2,4,8] Output: [1,2,4,8]

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 2 * 109 All the integers in nums are unique.

== Problem Number: 699 URL: https://leetcode.com/problems/sum-of-two-integers Title: 371. Sum of Two Integers Problem Description: Given two integers a and b, return the sum of the two integers without using the operators + and -. Example 1: Input: a = 1, b = 2 Output: 3 Example 2: Input: a = 2, b = 3 Output: 5

Constraints:

-1000 <= a, b <= 1000

==

Problem Number: 700 URL: https://leetcode.com/problems/super-pow Title: 372. Super Pow Problem Description: Your task is to calculate ab mod 1337 where a is a positive integer and b is an extremely large positive integer given in the form of an array. Example 1: Input: a = 2, b = [3] Output: 8

Example 2: Input: a = 2, b = [1,0] Output: 1024

Example 3: Input: a = 1, b = [4,3,3,8,5,2] Output: 1

Constraints:

1 <= a <= 231 - 1 1 <= b.length <= 2000 0 <= b[i] <= 9 b does not contain leading zeros.

==

Problem Number: 701 URL: https://leetcode.com/problems/find-k-pairs-with-smallest-sums Title: 373. Find K Pairs with Smallest Sums Problem Description: You are given two integer arrays nums1 and nums2 sorted in non-decreasing order and an integer k. Define a pair (u, v) which consists of one element from the first array and one element from the second array. Return the k pairs (u1, v1), (u2, v2), ..., (uk, vk) with the smallest sums. Example 1: Input: nums1 = [1,7,11], nums2 = [2,4,6], k = 3 Output: [[1,2],[1,4],[1,6]] Explanation: The first 3 pairs are returned from the sequence: [1,2],[1,4],[1,6],[7,2],[7,4],[11,2],[7,6],[11,4],[11,6]

Example 2: Input: nums1 = [1,1,2], nums2 = [1,2,3], k = 2 Output: [[1,1],[1,1]] Explanation: The first 2 pairs are returned from the sequence: [1,1],[1,1],[1,2],[2,1],[1,2],[2,2],[1,3],[1,3],[2,3]

Example 3: Input: nums1 = [1,2], nums2 = [3], k = 3 Output: [[1,3],[2,3]] Explanation: All possible pairs are returned from the sequence: [1,3],[2,3]

Constraints:

1 <= nums1.length, nums2.length <= 105 -109 <= nums1[i], nums2[i] <= 109 nums1 and nums2 both are sorted in non-decreasing order. 1 <= k <= 104

==

Problem Number: 702 URL: https://leetcode.com/problems/guess-number-higher-or-lower-ii Title: 375. Guess Number Higher or Lower II Problem Description: We are playing the Guessing Game. The game will work as follows:

I pick a number between 1 and n. You guess a number. If you guess the right number, you win the game. If you guess the wrong number, then I will tell you whether the number I picked is higher or lower, and you will continue guessing.

Every time you guess a wrong number x, you will pay x dollars. If you run out of money, you lose the game.

Given a particular n, return the minimum amount of money you need to guarantee a win regardless of what number I pick.   Example 1:

Input: n = 10 Output: 16 Explanation: The winning strategy is as follows: - The range is [1,10]. Guess 7.   - If this is my number, your total is $0. Otherwise, you pay $7.   - If my number is higher, the range is [8,10]. Guess 9.   - If this is my number, your total is $7. Otherwise, you pay $9.   - If my number is higher, it must be 10. Guess 10. Your total is $7 + $9 = $16.   - If my number is lower, it must be 8. Guess 8. Your total is $7 + $9 = $16.   - If my number is lower, the range is [1,6]. Guess 3.   - If this is my number, your total is $7. Otherwise, you pay $3.   - If my number is higher, the range is [4,6]. Guess 5.   - If this is my number, your total is $7 + $3 = $10. Otherwise, you pay $5.   - If my number is higher, it must be 6. Guess 6. Your total is $7 + $3 + $5 = $15.   - If my number is lower, it must be 4. Guess 4. Your total is $7 + $3 + $5 = $15.   - If my number is lower, the range is [1,2]. Guess 1.   - If this is my number, your total is $7 + $3 = $10. Otherwise, you pay $1.   - If my number is higher, it must be 2. Guess 2. Your total is $7 + $3 + $1 = $11. The worst case in all these scenarios is that you pay $16. Hence, you only need $16 to guarantee a win.

Example 2: Input: n = 1 Output: 0 Explanation: There is only one possible number, so you can guess 1 and not have to pay anything.

Example 3: Input: n = 2 Output: 1 Explanation: There are two possible numbers, 1 and 2. - Guess 1.   - If this is my number, your total is $0. Otherwise, you pay $1.   - If my number is higher, it must be 2. Guess 2. Your total is $1. The worst case is that you pay $1.

Constraints:

1 <= n <= 200

==

Problem Number: 703 URL: https://leetcode.com/problems/wiggle-subsequence Title: 376. Wiggle Subsequence Problem Description: A wiggle sequence is a sequence where the differences between successive numbers strictly alternate between positive and negative. The first difference (if one exists) may be either positive or negative. A sequence with one element and a sequence with two non-equal elements are trivially wiggle sequences.

For example, [1, 7, 4, 9, 2, 5] is a wiggle sequence because the differences (6, -3, 5, -7, 3) alternate between positive and negative. In contrast, [1, 4, 7, 2, 5] and [1, 7, 4, 5, 5] are not wiggle sequences. The first is not because its first two differences are positive, and the second is not because its last difference is zero.

A subsequence is obtained by deleting some elements (possibly zero) from the original sequence, leaving the remaining elements in their original order. Given an integer array nums, return the length of the longest wiggle subsequence of nums. Example 1: Input: nums = [1,7,4,9,2,5] Output: 6 Explanation: The entire sequence is a wiggle sequence with differences (6, -3, 5, -7, 3).

Example 2: Input: nums = [1,17,5,10,13,15,10,5,16,8] Output: 7 Explanation: There are several subsequences that achieve this length. One is [1, 17, 10, 13, 10, 16, 8] with differences (16, -7, 3, -3, 6, -8).

Example 3: Input: nums = [1,2,3,4,5,6,7,8,9] Output: 2

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] <= 1000

Follow up: Could you solve this in O(n) time?

==

Problem Number: 704 URL: https://leetcode.com/problems/combination-sum-iv Title: 377. Combination Sum IV Problem Description: Given an array of distinct integers nums and a target integer target, return the number of possible combinations that add up to target. The test cases are generated so that the answer can fit in a 32-bit integer. Example 1: Input: nums = [1,2,3], target = 4 Output: 7 Explanation: The possible combination ways are: (1, 1, 1, 1) (1, 1, 2) (1, 2, 1) (1, 3) (2, 1, 1) (2, 2) (3, 1) Note that different sequences are counted as different combinations.

Example 2: Input: nums = [9], target = 3 Output: 0

Constraints:

1 <= nums.length <= 200 1 <= nums[i] <= 1000 All the elements of nums are unique. 1 <= target <= 1000

Follow up: What if negative numbers are allowed in the given array? How does it change the problem? What limitation we need to add to the question to allow negative numbers?

==

Problem Number: 705 URL: https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix Title: 378. Kth Smallest Element in a Sorted Matrix Problem Description: Given an n x n matrix where each of the rows and columns is sorted in ascending order, return the kth smallest element in the matrix. Note that it is the kth smallest element in the sorted order, not the kth distinct element. You must find a solution with a memory complexity better than O(n2). Example 1: Input: matrix = [[1,5,9],[10,11,13],[12,13,15]], k = 8 Output: 13 Explanation: The elements in the matrix are [1,5,9,10,11,12,13,13,15], and the 8th smallest number is 13

Example 2: Input: matrix = [[-5]], k = 1 Output: -5

Constraints:

n == matrix.length == matrix[i].length 1 <= n <= 300 -109 <= matrix[i][j] <= 109 All the rows and columns of matrix are guaranteed to be sorted in non-decreasing order. 1 <= k <= n2

Follow up:

Could you solve the problem with a constant memory (i.e., O(1) memory complexity)? Could you solve the problem in O(n) time complexity? The solution may be too advanced for an interview but you may find reading this paper fun.

==
Problem Number: 706 URL: https://leetcode.com/problems/insert-delete-getrandom-o1 Title: 380. Insert Delete GetRandom O(1) Problem Description: Implement the RandomizedSet class:

RandomizedSet() Initializes the RandomizedSet object. bool insert(int val) Inserts an item val into the set if not present. Returns true if the item was not present, false otherwise. bool remove(int val) Removes an item val from the set if present. Returns true if the item was present, false otherwise. int getRandom() Returns a random element from the current set of elements (it's guaranteed that at least one element exists when this method is called). Each element must have the same probability of being returned.

You must implement the functions of the class such that each function works in average O(1) time complexity. Example 1: Input ["RandomizedSet", "insert", "remove", "insert", "getRandom", "remove", "insert", "getRandom"] [[], [1], [2], [2], [], [1], [2], []] Output [null, true, false, true, 2, true, false, 2]

Explanation RandomizedSet randomizedSet = new RandomizedSet(); randomizedSet.insert(1); // Inserts 1 to the set. Returns true as 1 was inserted successfully. randomizedSet.remove(2); // Returns false as 2 does not exist in the set. randomizedSet.insert(2); // Inserts 2 to the set, returns true. Set now contains [1,2]. randomizedSet.getRandom(); // getRandom() should return either 1 or 2 randomly. randomizedSet.remove(1); // Removes 1 from the set, returns true. Set now contains [2]. randomizedSet.insert(2); // 2 was already in the set, so return false. randomizedSet.getRandom(); // Since 2 is the only number in the set, getRandom() will always return 2.

Constraints:

-231 <= val <= 231 - 1 At most 2 * 105 calls will be made to insert, remove, and getRandom. There will be at least one element in the data structure when getRandom is called.

==
Problem Number: 707 URL: https://leetcode.com/problems/linked-list-random-node Title: 382. Linked List Random Node Problem Description: Given a singly linked list, return a random node's value from the linked list. Each node must have the same probability of being chosen. Implement the Solution class:

Solution(ListNode head) Initializes the object with the head of the singly-linked list head. int getRandom() Chooses a node randomly from the list and returns its value. All the nodes of the list should be equally likely to be chosen.

Example 1:

Input ["Solution", "getRandom", "getRandom", "getRandom", "getRandom", "getRandom"] [[[1, 2, 3]], [], [], [], [], []] Output [null, 1, 3, 2, 2, 3]

Explanation Solution solution = new Solution([1, 2, 3]); solution.getRandom(); // return 1 solution.getRandom(); // return 3 solution.getRandom(); // return 2 solution.getRandom(); // return 2 solution.getRandom(); // return 3 // getRandom() should return either 1, 2, or 3 randomly. Each element should have equal probability of returning.

Constraints:

The number of nodes in the linked list will be in the range [1, 104]. -104 <= Node.val <= 104 At most 104 calls will be made to getRandom.

Follow up:

What if the linked list is extremely large and its length is unknown to you? Could you solve this efficiently without using extra space?

==
Problem Number: 708 URL: https://leetcode.com/problems/shuffle-an-array Title: 384. Shuffle an Array Problem Description: Given an integer array nums, design an algorithm to randomly shuffle the array. All permutations of the array should be equally likely as a result of the shuffling. Implement the Solution class:

Solution(int[] nums) Initializes the object with the integer array nums. int[] reset() Resets the array to its original configuration and returns it. int[] shuffle() Returns a random shuffling of the array.

Example 1: Input ["Solution", "shuffle", "reset", "shuffle"] [[[1, 2, 3]], [], [], []] Output [null, [3, 1, 2], [1, 2, 3], [1, 3, 2]]

Explanation Solution solution = new Solution([1, 2, 3]); solution.shuffle(); // Shuffle the array [1,2,3] and return its result. // Any permutation of [1,2,3] must be equally likely to be returned. // Example: return [3, 1, 2] solution.reset(); // Resets the array back to its original configuration [1,2,3]. Return [1, 2, 3] solution.shuffle(); // Returns the random shuffling of array [1,2,3]. Example: return [1, 3, 2]

Constraints:

1 <= nums.length <= 50 -106 <= nums[i] <= 106 All the elements of nums are unique. At most 104 calls in total will be made to reset and shuffle.

==
Problem Number: 709 URL: https://leetcode.com/problems/mini-parser Title:

385. Mini Parser Problem Description: Given a string s represents the serialization of a nested list, implement a parser to deserialize it and return the deserialized NestedInteger. Each element is either an integer or a list whose elements may also be integers or other lists. Example 1: Input: s = "324" Output: 324 Explanation: You should return a NestedInteger object which contains a single integer 324.

Example 2: Input: s = "[123,[456,[789]]]" Output: [123,[456,[789]]] Explanation: Return a NestedInteger object containing a nested list with 2 elements: 1. An integer containing value 123. 2. A nested list containing two elements: i. An integer containing value 456. ii. A nested list with one element: a. An integer containing value 789

Constraints:

1 <= s.length <= 5 * 104 s consists of digits, square brackets "[]", negative sign '-', and commas ','. s is the serialization of valid NestedInteger. All the values in the input are in the range [-106, 106].

==
Problem Number: 710 URL: https://leetcode.com/problems/lexicographical-numbers Title: 386. Lexicographical Numbers Problem Description: Given an integer n, return all the numbers in the range [1, n] sorted in lexicographical order. You must write an algorithm that runs in O(n) time and uses O(1) extra space. Example 1: Input: n = 13 Output: [1,10,11,12,13,2,3,4,5,6,7,8,9] Example 2: Input: n = 2 Output: [1,2]

Constraints:

1 <= n <= 5 * 104

==
Problem Number: 711 URL: https://leetcode.com/problems/longest-absolute-file-path Title: 388. Longest Absolute File Path Problem Description: Suppose we have a file system that stores both files and directories. An example of one system is represented in the following picture:

Here, we have dir as the only directory in the root. dir contains two subdirectories, subdir1 and subdir2. subdir1 contains a file file1.ext and subdirectory subsubdir1. subdir2 contains a subdirectory subsubdir2, which contains a file file2.ext. In text form, it looks like this (with representing the tab character): dir subdir1 file1.ext subsubdir1 subdir2 subsubdir2 file2.ext

If we were to write this representation in code, it will look like this: "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext". Note that the '\n' and '\t' are the new-line and tab characters. Every file and directory has a unique absolute path in the file system, which is the order of directories that must be opened to reach the file/directory itself, all concatenated by '/'s. Using the above example, the absolute path to file2.ext is "dir/subdir2/subsubdir2/file2.ext". Each directory name consists of letters, digits, and/or spaces. Each file name is of the form name.extension, where name and extension consist of letters, digits, and/or spaces. Given a string input representing the file system in the explained format, return the length of the longest absolute path to a file in the abstracted file system. If there is no file in the system, return 0. Note that the testcases are generated such that the file system is valid and no file or directory name has length 0. Example 1:

Input: input = "dir\n\tsubdir1\n\tsubdir2\n\t\tfile.ext" Output: 20 Explanation: We have only one file, and the absolute path is "dir/subdir2/file.ext" of length 20.

Example 2:

Input: input = "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext" Output: 32 Explanation: We have two files: "dir/subdir1/file1.ext" of length 21 "dir/subdir2/subsubdir2/file2.ext" of length 32. We return 32 since it is the longest absolute path to a file.

Example 3: Input: input = "a" Output: 0 Explanation: We do not have any files, just a single directory named "a".

Constraints:

1 <= input.length <= 104 input may contain lowercase or uppercase English letters, a new line character '\n', a tab character '\t', a dot '.', a space ' ', and digits. All file and directory names have positive length.

==
Problem Number: 712 URL: https://leetcode.com/problems/elimination-game Title: 390. Elimination Game Problem Description: You have a list arr of all integers in the range [1, n] sorted in a strictly increasing order. Apply the following algorithm on arr:

Starting from left to right, remove the first number and every other number afterward until you reach the end of the list. Repeat the previous step again, but this time from right to left, remove the rightmost number and every other number from the remaining numbers. Keep repeating the steps again, alternating left to right and right to left, until a single number remains.

Given the integer n, return the last number that remains in arr. Example 1: Input: n = 9 Output: 6 Explanation: arr = [1, 2, 3, 4, 5, 6, 7, 8, 9] arr = [2, 4, 6, 8] arr = [2, 6] arr = [6]

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 109

==
Problem Number: 713 URL: https://leetcode.com/problems/utf-8-validation Title: 393. UTF-8 Validation Problem Description: Given an integer array

data representing the data, return whether it is a valid UTF-8 encoding (i.e. it translates to a sequence of valid UTF-8 encoded characters). A character in UTF8 can be from 1 to 4 bytes long, subjected to the following rules:

For a 1-byte character, the first bit is a 0, followed by its Unicode code. For an n-bytes character, the first n bits are all one's, the n + 1 bit is 0, followed by n - 1 bytes with the most significant 2 bits being 10.

This is how the UTF-8 encoding would work: Number of Bytes | UTF-8 Octet Sequence | (binary) --------------------+--- 1 | 0xxxxxxx 2 | 110xxxxx 10xxxxxx 3 | 1110xxxx 10xxxxxx 10xxxxxx 4 | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

x denotes a bit in the binary form of a byte that may be either 0 or 1. Note: The input is an array of integers. Only the least significant 8 bits of each integer is used to store the data. This means each integer represents only 1 byte of data.   Example 1: Input: data = [197,130,1] Output: true Explanation: data represents the octet sequence: 11000101 10000010 00000001. It is a valid utf-8 encoding for a 2-bytes character followed by a 1-byte character.

Example 2: Input: data = [235,140,4] Output: false Explanation: data represented the octet sequence: 11101011 10001100 00000100. The first 3 bits are all one's and the 4th bit is 0 means it is a 3-bytes character. The next byte is a continuation byte which starts with 10 and that's correct. But the second continuation byte does not start with 10, so it is invalid.

Constraints:

1 <= data.length <= 2 * 104 0 <= data[i] <= 255

==

Problem Number: 714 URL: https://leetcode.com/problems/decode-string Title: 394. Decode String Problem Description: Given an encoded string, return its decoded string. The encoding rule is: k[encoded_string], where the encoded_string inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer. You may assume that the input string is always valid; there are no extra white spaces, square brackets are well-formed, etc. Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k. For example, there will not be input like 3a or 2[4]. The test cases are generated so that the length of the output will never exceed 105.   Example 1: Input: s = "3[a]2[bc]" Output: "aaabcbc"

Example 2: Input: s = "3[a2[c]]" Output: "accaccacc"

Example 3: Input: s = "2[abc]3[cd]ef" Output: "abcabccdcdcdef"

Constraints:

1 <= s.length <= 30 s consists of lowercase English letters, digits, and square brackets '[]'. s is guaranteed to be a valid input. All the integers in s are in the

range [1, 300].

==
Problem Number: 715 URL: https://leetcode.com/problems/longest-substring-with-at-least-k-repeating-characters Title: 395. Longest Substring with At Least K Repeating Characters Problem Description: Given a string s and an integer k, return the length of the longest substring of s such that the frequency of each character in this substring is greater than or equal to k. if no such substring exists, return 0. Example 1: Input: s = "aaabb", k = 3 Output: 3 Explanation: The longest substring is "aaa", as 'a' is repeated 3 times.

Example 2: Input: s = "ababbc", k = 2 Output: 5 Explanation: The longest substring is "ababb", as 'a' is repeated 2 times and 'b' is repeated 3 times.

Constraints:

1 <= s.length <= 104 s consists of only lowercase English letters. 1 <= k <= 105

==
Problem Number: 716 URL: https://leetcode.com/problems/rotate-function Title: 396. Rotate Function Problem Description: You are given an integer array nums of length n. Assume arrk to be an array obtained by rotating nums by k positions clock-wise. We define the rotation function F on nums as follow:

F(k) = 0 * arrk[0] + 1 * arrk[1] + ... + (n - 1) * arrk[n - 1].

Return the maximum value of F(0), F(1), ..., F(n-1). The test cases are generated so that the answer fits in a 32-bit integer. Example 1: Input: nums = [4,3,2,6] Output: 26 Explanation: F(0) = (0 * 4) + (1 * 3) + (2 * 2) + (3 * 6) = 0 + 3 + 4 + 18 = 25 F(1) = (0 * 6) + (1 * 4) + (2 * 3) + (3 * 2) = 0 + 4 + 6 + 6 = 16 F(2) = (0 * 2) + (1 * 6) + (2 * 4) + (3 * 3) = 0 + 6 + 8 + 9 = 23 F(3) = (0 * 3) + (1 * 2) + (2 * 6) + (3 * 4) = 0 + 2 + 12 + 12 = 26 So the maximum value of F(0), F(1), F(2), F(3) is F(3) = 26.

Example 2: Input: nums = [100] Output: 0

Constraints:

n == nums.length 1 <= n <= 105 -100 <= nums[i] <= 100

==
Problem Number: 717 URL: https://leetcode.com/problems/integer-replacement Title: 397. Integer Replacement Problem Description: Given a positive integer n, you can apply one of the following operations:

If n is even, replace n with n / 2. If n is odd, replace n with either n + 1 or n - 1.

Return the minimum number of operations needed for n to become 1. Example 1: Input: n = 8 Output: 3 Explanation: 8 -> 4 -> 2 -> 1

Example 2: Input: n = 7 Output: 4 Explanation: 7 -> 8 -> 4 -> 2 -> 1 or 7 -> 6 -> 3 -> 2 -> 1

Example 3: Input: n = 4 Output: 2

Constraints:

1 <= n <= 231 - 1

==

Problem Number: 718 URL: https://leetcode.com/problems/random-pick-index Title: 398. Random Pick Index Problem Description: Given an integer array nums with possible duplicates, randomly output the index of a given target number. You can assume that the given target number must exist in the array. Implement the Solution class:

Solution(int[] nums) Initializes the object with the array nums. int pick(int target) Picks a random index i from nums where nums[i] == target. If there are multiple valid i's, then each index should have an equal probability of returning.

Example 1: Input ["Solution", "pick", "pick", "pick"] [[[1, 2, 3, 3, 3]], [3], [1], [3]] Output [null, 4, 0, 2]

Explanation Solution solution = new Solution([1, 2, 3, 3, 3]); solution.pick(3); // It should return either index 2, 3, or 4 randomly. Each index should have equal probability of returning. solution.pick(1); // It should return 0. Since in the array only nums[0] is equal to 1. solution.pick(3); // It should return either index 2, 3, or 4 randomly. Each index should have equal probability of returning.

Constraints:

1 <= nums.length <= 2 * 104 -231 <= nums[i] <= 231 - 1 target is an integer from nums. At most 104 calls will be made to pick.

==

Problem Number: 719 URL: https://leetcode.com/problems/evaluate-division Title: 399. Evaluate Division Problem Description: You are given an array of variable pairs equations and an array of real numbers values, where equations[i] = [Ai, Bi] and values[i] represent the equation Ai / Bi = values[i]. Each Ai or Bi is a string that represents a single variable. You are also given some queries, where queries[j] = [Cj, Dj] represents the jth query where you must find the answer for Cj / Dj = ?. Return the answers to all queries. If a single answer cannot be determined, return -1.0. Note: The input is always valid. You may assume that evaluating the queries will not result in division by zero and that there is no contradiction. Note: The variables that do not occur in the list of equations are undefined, so the answer cannot be determined for them. Example 1: Input: equations = [["a","b"],["b","c"]], values = [2.0,3.0], queries = [["a","c"],["b","a"],["a","e"],["a","a"],["x","x"]] Output: [6.00000,0.50000,-1.00000,1.00000,-1.00000] Explanation: Given: a / b = 2.0, b / c = 3.0 queries are: a / c = ?, b / a = ?, a / e = ?, a / a

= ?, x / x = ? return: [6.0, 0.5, -1.0, 1.0, -1.0] note: x is undefined => -1.0 Example 2: Input: equations = [["a","b"],["b","c"],["bc","cd"]], values = [1.5,2.5,5.0], queries = [["a","c"],["c","b"],["bc","cd"],["cd","bc"]] Output: [3.75000,0.40000,5.00000,0.20000]

Example 3: Input: equations = [["a","b"]], values = [0.5], queries = [["a","b"],["b","a"],["a","c"],["x","y"]] Output: [0.50000,2.00000,-1.00000,-1.00000]

Constraints:

1 <= equations.length <= 20 equations[i].length == 2 1 <= Ai.length, Bi.length <= 5 values.length == equations.length 0.0 < values[i] <= 20.0 1 <= queries.length <= 20 queries[i].length == 2 1 <= Cj.length, Dj.length <= 5 Ai, Bi, Cj, Dj consist of lower case English letters and digits.

==
Problem Number: 720 URL: https://leetcode.com/problems/nth-digit Title: 400. Nth Digit Problem Description: Given an integer n, return the nth digit of the infinite integer sequence [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...]. Example 1: Input: n = 3 Output: 3

Example 2: Input: n = 11 Output: 0 Explanation: The 11th digit of the sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... is a 0, which is part of the number 10.

Constraints:

1 <= n <= 231 - 1

==
Problem Number: 721 URL: https://leetcode.com/problems/remove-k-digits Title: 402. Remove K Digits Problem Description: Given string num representing a non-negative integer num, and an integer k, return the smallest possible integer after removing k digits from num. Example 1: Input: num = "1432219", k = 3 Output: "1219" Explanation: Remove the three digits 4, 3, and 2 to form the new number 1219 which is the smallest.

Example 2: Input: num = "10200", k = 1 Output: "200" Explanation: Remove the leading 1 and the number is 200. Note that the output must not contain leading zeroes.

Example 3: Input: num = "10", k = 2 Output: "0" Explanation: Remove all the digits from the number and it is left with nothing which is 0.

Constraints:

1 <= k <= num.length <= 105 num consists of only digits. num does not have any leading zeros except for the zero itself.

==
Problem Number: 722 URL: https://leetcode.com/problems/queue-reconstruction-

by-height Title: 406. Queue Reconstruction by Height Problem Description: You are given an array of people, people, which are the attributes of some people in a queue (not necessarily in order). Each people[i] = [hi, ki] represents the ith person of height hi with exactly ki other people in front who have a height greater than or equal to hi. Reconstruct and return the queue that is represented by the input array people. The returned queue should be formatted as an array queue, where queue[j] = [hj, kj] is the attributes of the jth person in the queue (queue[0] is the person at the front of the queue).   Example 1: Input: people = [[7,0],[4,4],[7,1],[5,0],[6,1],[5,2]] Output: [[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]] Explanation: Person 0 has height 5 with no other people taller or the same height in front. Person 1 has height 7 with no other people taller or the same height in front. Person 2 has height 5 with two persons taller or the same height in front, which is person 0 and 1. Person 3 has height 6 with one person taller or the same height in front, which is person 1. Person 4 has height 4 with four people taller or the same height in front, which are people 0, 1, 2, and 3. Person 5 has height 7 with one person taller or the same height in front, which is person 1. Hence [[5,0],[7,0],[5,2],[6,1],[4,4],[7,1]] is the reconstructed queue.

Example 2: Input: people = [[6,0],[5,0],[4,0],[3,2],[2,2],[1,4]] Output: [[4,0],[5,0],[2,2],[3,2],[1,4],[6,0]]

Constraints:

1 <= people.length <= 2000 0 <= hi <= 106 0 <= ki < people.length It is guaranteed that the queue can be reconstructed.

==
Problem Number: 723 URL: https://leetcode.com/problems/arithmetic-slices Title: 413. Arithmetic Slices Problem Description: An integer array is called arithmetic if it consists of at least three elements and if the difference between any two consecutive elements is the same.

For example, [1,3,5,7,9], [7,7,7,7], and [3,-1,-5,-9] are arithmetic sequences.

Given an integer array nums, return the number of arithmetic subarrays of nums. A subarray is a contiguous subsequence of the array.   Example 1: Input: nums = [1,2,3,4] Output: 3 Explanation: We have 3 arithmetic slices in nums: [1, 2, 3], [2, 3, 4] and [1,2,3,4] itself.

Example 2: Input: nums = [1] Output: 0

Constraints:

1 <= nums.length <= 5000 -1000 <= nums[i] <= 1000

==
Problem Number: 724 URL: https://leetcode.com/problems/partition-equal-subset-sum Title: 416. Partition Equal Subset Sum Problem Description: Given an integer array nums, return true if you can partition the array into two subsets such that the sum of the elements in both subsets is equal or false

otherwise. Example 1: Input: nums = [1,5,11,5] Output: true Explanation: The array can be partitioned as [1, 5, 5] and [11].

Example 2: Input: nums = [1,2,3,5] Output: false Explanation: The array cannot be partitioned into equal sum subsets.

Constraints:

1 <= nums.length <= 200 1 <= nums[i] <= 100

==
Problem Number: 725 URL: https://leetcode.com/problems/pacific-atlantic-water-flow Title: 417. Pacific Atlantic Water Flow Problem Description: There is an m x n rectangular island that borders both the Pacific Ocean and Atlantic Ocean. The Pacific Ocean touches the island's left and top edges, and the Atlantic Ocean touches the island's right and bottom edges. The island is partitioned into a grid of square cells. You are given an m x n integer matrix heights where heights[r][c] represents the height above sea level of the cell at coordinate (r, c). The island receives a lot of rain, and the rain water can flow to neighboring cells directly north, south, east, and west if the neighboring cell's height is less than or equal to the current cell's height. Water can flow from any cell adjacent to an ocean into the ocean. Return a 2D list of grid coordinates result where result[i] = [ri, ci] denotes that rain water can flow from cell (ri, ci) to both the Pacific and Atlantic oceans. Example 1:

Input: heights = [[1,2,2,3,5],[3,2,3,4,4],[2,4,5,3,1],[6,7,1,4,5],[5,1,1,2,4]] Output: [[0,4],[1,3],[1,4],[2,2],[3,0],[3,1],[4,0]] Explanation: The following cells can flow to the Pacific and Atlantic oceans, as shown below: [0,4]: [0,4] -> Pacific Ocean [0,4] -> Atlantic Ocean [1,3]: [1,3] -> [0,3] -> Pacific Ocean [1,3] -> [1,4] -> Atlantic Ocean [1,4]: [1,4] -> [1,3] -> [0,3] -> Pacific Ocean [1,4] -> Atlantic Ocean [2,2]: [2,2] -> [1,2] -> [0,2] -> Pacific Ocean [2,2] -> [2,3] -> [2,4] -> Atlantic Ocean [3,0]: [3,0] -> Pacific Ocean [3,0] -> [4,0] -> Atlantic Ocean [3,1]: [3,1] -> [3,0] -> Pacific Ocean [3,1] -> [4,1] -> Atlantic Ocean [4,0]: [4,0] -> Pacific Ocean [4,0] -> Atlantic Ocean Note that there are other possible paths for these cells to flow to the Pacific and Atlantic oceans.

Example 2: Input: heights = [[1]] Output: [[0,0]] Explanation: The water can flow from the only cell to the Pacific and Atlantic oceans.

Constraints:

m == heights.length n == heights[r].length 1 <= m, n <= 200 0 <= heights[r][c] <= 105

==
Problem Number: 726 URL: https://leetcode.com/problems/battleships-in-a-board Title: 419. Battleships in a Board Problem Description: Given an m x n matrix board where each cell is a battleship 'X' or empty '.', return the number of the battleships on board. Battleships can only be placed horizontally or vertically on board. In other words, they can only be made of the shape 1 x

k (1 row, k columns) or k x 1 (k rows, 1 column), where k can be of any size. At least one horizontal or vertical cell separates between two battleships (i.e., there are no adjacent battleships). Example 1:

Input: board = [["X",".",".","X"],[".",".",".","X"],[".",".",".","X"]] Output: 2

Example 2: Input: board = [["."]] Output: 0

Constraints:

m == board.length n == board[i].length 1 <= m, n <= 200 board[i][j] is either '.' or 'X'.

Follow up: Could you do it in one-pass, using only O(1) extra memory and without modifying the values board?

==
Problem Number: 727 URL: https://leetcode.com/problems/maximum-xor-of-two-numbers-in-an-array Title: 421. Maximum XOR of Two Numbers in an Array Problem Description: Given an integer array nums, return the maximum result of nums[i] XOR nums[j], where 0 <= i <= j < n. Example 1: Input: nums = [3,10,5,25,2,8] Output: 28 Explanation: The maximum result is 5 XOR 25 = 28.

Example 2: Input: nums = [14,70,53,83,49,91,36,80,92,51,66,70] Output: 127

Constraints:

1 <= nums.length <= 2 * 105 0 <= nums[i] <= 231 - 1

==
Problem Number: 728 URL: https://leetcode.com/problems/reconstruct-original-digits-from-english Title: 423. Reconstruct Original Digits from English Problem Description: Given a string s containing an out-of-order English representation of digits 0-9, return the digits in ascending order. Example 1: Input: s = "owoztneoer" Output: "012" Example 2: Input: s = "fviefuro" Output: "45"

Constraints:

1 <= s.length <= 105 s[i] is one of the characters ["e","g","f","i","h","o","n","s","r","u","t","w","v","x","z"]. s is guaranteed to be valid.

==
Problem Number: 729 URL: https://leetcode.com/problems/longest-repeating-character-replacement Title: 424. Longest Repeating Character Replacement Problem Description: You are given a string s and an integer k. You can choose any character of the string and change it to any other uppercase English character. You can perform this operation at most k times. Return the length of the longest substring containing the same letter you can get after performing the above operations. Example 1: Input: s = "ABAB", k = 2 Output: 4 Explanation: Replace the two 'A's with two 'B's or vice versa.

Example 2: Input: s = "AABABBA", k = 1 Output: 4 Explanation: Replace the one 'A' in the middle with 'B' and form "AABBBBA". The substring "BBBB" has the longest repeating letters, which is 4. There may exists other ways to achive this answer too.   Constraints:

1 <= s.length <= 105 s consists of only uppercase English letters. 0 <= k <= s.length

==
Problem Number: 730 URL: https://leetcode.com/problems/construct-quad-tree Title: 427. Construct Quad Tree Problem Description: Given a n * n matrix grid of 0's and 1's only. We want to represent grid with a Quad-Tree. Return the root of the Quad-Tree representing grid. A Quad-Tree is a tree data structure in which each internal node has exactly four children. Besides, each node has two attributes:

val: True if the node represents a grid of 1's or False if the node represents a grid of 0's. Notice that you can assign the val to True or False when isLeaf is False, and both are accepted in the answer. isLeaf: True if the node is a leaf node on the tree or False if the node has four children.

class Node { public boolean val; public boolean isLeaf; public Node topLeft; public Node topRight; public Node bottomLeft; public Node bottomRight; } We can construct a Quad-Tree from a two-dimensional area using the following steps:

If the current grid has the same value (i.e all 1's or all 0's) set isLeaf True and set val to the value of the grid and set the four children to Null and stop. If the current grid has different values, set isLeaf to False and set val to any value and divide the current grid into four sub-grids as shown in the photo. Recurse for each of the children with the proper sub-grid.

If you want to know more about the Quad-Tree, you can refer to the wiki. Quad-Tree format: You don't need to read this section for solving the problem. This is only if you want to understand the output format here. The output represents the serialized format of a Quad-Tree using level order traversal, where null signifies a path terminator where no node exists below. It is very similar to the serialization of the binary tree. The only difference is that the node is represented as a list [isLeaf, val]. If the value of isLeaf or val is True we represent it as 1 in the list [isLeaf, val] and if the value of isLeaf or val is False we represent it as 0.   Example 1:

Input: grid = [[0,1],[1,0]] Output: [[0,1],[1,0],[1,1],[1,1],[1,0]] Explanation: The explanation of this example is shown below: Notice that 0 represents False and 1 represents True in the photo representing the Quad-Tree.

Example 2:

Input: grid = [[1,1,1,1,0,0,0,0],[1,1,1,1,0,0,0,0],[1,1,1,1,1,1,1,1],[1,1,1,1,1,1,1,1],[1,1,1,1,0,0,0,0],[1,1,1,1,0,0,0,0],[1,
Output: [[0,1],[1,1],[0,1],[1,1],[1,0],null,null,null,null,[1,0],[1,0],[1,1],[1,1]] Expla-

nation: All values in the grid are not the same. We divide the grid into four sub-grids. The topLeft, bottomLeft and bottomRight each has the same value. The topRight have different values so we divide it into 4 sub-grids where each has the same value. Explanation is shown in the photo below:

Constraints:

n == grid.length == grid[i].length n == 2x where 0 <= x <= 6

==
Problem Number: 731 URL: https://leetcode.com/problems/n-ary-tree-level-order-traversal Title: 429. N-ary Tree Level Order Traversal Problem Description: Given an n-ary tree, return the level order traversal of its nodes' values. Nary-Tree input serialization is represented in their level order traversal, each group of children is separated by the null value (See examples). Example 1:

Input: root = [1,null,3,2,4,null,5,6] Output: [[1],[3,2,4],[5,6]]

Example 2:

Input: root = [1,null,2,3,4,5,null,null,6,7,null,8,null,9,10,null,null,11,null,12,null,13,null,null,14] Output: [[1],[2,3,4,5],[6,7,8,9,10],[11,12,13],[14]]

Constraints:

The height of the n-ary tree is less than or equal to 1000 The total number of nodes is between [0, 104]

==
Problem Number: 732 URL: https://leetcode.com/problems/flatten-a-multilevel-doubly-linked-list Title: 430. Flatten a Multilevel Doubly Linked List Problem Description: You are given a doubly linked list, which contains nodes that have a next pointer, a previous pointer, and an additional child pointer. This child pointer may or may not point to a separate doubly linked list, also containing these special nodes. These child lists may have one or more children of their own, and so on, to produce a multilevel data structure as shown in the example below. Given the head of the first level of the list, flatten the list so that all the nodes appear in a single-level, doubly linked list. Let curr be a node with a child list. The nodes in the child list should appear after curr and before curr.next in the flattened list. Return the head of the flattened list. The nodes in the list must have all of their child pointers set to null. Example 1:

Input: head = [1,2,3,4,5,6,null,null,null,7,8,9,10,null,null,11,12] Output: [1,2,3,7,8,11,12,9,10,4,5,6] Explanation: The multilevel linked list in the input is shown. After flattening the multilevel linked list it becomes:

Example 2:

Input: head = [1,2,null,3] Output: [1,3,2] Explanation: The multilevel linked list in the input is shown. After flattening the multilevel linked list it becomes:

Example 3: Input: head = [] Output: [] Explanation: There could be empty list in the input.

Constraints:

The number of Nodes will not exceed 1000. 1 <= Node.val <= 105

How the multilevel linked list is represented in test cases: We use the multilevel linked list from Example 1 above: 1---2---3---4---5---6--NULL | 7---8---9---10--NULL | 11--12--NULL The serialization of each level is as follows: [1,2,3,4,5,6,null] [7,8,9,10,null] [11,12,null]

To serialize all levels together, we will add nulls in each level to signify no node connects to the upper node of the previous level. The serialization becomes: [1, 2, 3, 4, 5, 6, null] | [null, null, 7, 8, 9, 10, null] | [null, 11, 12, null]

Merging the serialization of each level and removing trailing nulls we obtain: [1,2,3,4,5,6,null,null,null,7,8,9,10,null,null,11,12]

==

Problem Number: 733 URL: https://leetcode.com/problems/minimum-genetic-mutation Title: 433. Minimum Genetic Mutation Problem Description: A gene string can be represented by an 8-character long string, with choices from 'A', 'C', 'G', and 'T'. Suppose we need to investigate a mutation from a gene string startGene to a gene string endGene where one mutation is defined as one single character changed in the gene string.

For example, "AACCGGTT" --> "AACCGGTA" is one mutation.

There is also a gene bank bank that records all the valid gene mutations. A gene must be in bank to make it a valid gene string. Given the two gene strings startGene and endGene and the gene bank bank, return the minimum number of mutations needed to mutate from startGene to endGene. If there is no such a mutation, return -1. Note that the starting point is assumed to be valid, so it might not be included in the bank. Example 1: Input: startGene = "AACCGGTT", endGene = "AACCGGTA", bank = ["AACCGGTA"] Output: 1

Example 2: Input: startGene = "AACCGGTT", endGene = "AAACGGTA", bank = ["AACCGGTA","AACCGCTA","AAACGGTA"] Output: 2

Constraints:

0 <= bank.length <= 10 startGene.length == endGene.length == bank[i].length == 8 startGene, endGene, and bank[i] consist of only the characters ['A', 'C', 'G', 'T'].

==

Problem Number: 734 URL: https://leetcode.com/problems/non-overlapping-intervals Title: 435. Non-overlapping Intervals Problem Description: Given an array of intervals intervals where intervals[i] = [starti, endi], return the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.   Example 1: Input: intervals = [[1,2],[2,3],[3,4],[1,3]] Output: 1 Explanation: [1,3] can be removed and the rest of the intervals are non-overlapping.

Example 2: Input: intervals = [[1,2],[1,2],[1,2]] Output: 2 Explanation: You need to remove two [1,2] to make the rest of the intervals non-overlapping.

Example 3: Input: intervals = [[1,2],[2,3]] Output: 0 Explanation: You don't need to remove any of the intervals since they're already non-overlapping.

Constraints:

1 <= intervals.length <= 105 intervals[i].length == 2 -5 * 104 <= starti < endi <= 5 * 104

==
Problem Number: 735 URL: https://leetcode.com/problems/find-right-interval Title: 436. Find Right Interval Problem Description: You are given an array of intervals, where intervals[i] = [starti, endi] and each starti is unique. The right interval for an interval i is an interval j such that startj >= endi and startj is minimized. Note that i may equal j. Return an array of right interval indices for each interval i. If no right interval exists for interval i, then put -1 at index i.   Example 1: Input: intervals = [[1,2]] Output: [-1] Explanation: There is only one interval in the collection, so it outputs -1.

Example 2: Input: intervals = [[3,4],[2,3],[1,2]] Output: [-1,0,1] Explanation: There is no right interval for [3,4]. The right interval for [2,3] is [3,4] since start0 = 3 is the smallest start that is >= end1 = 3. The right interval for [1,2] is [2,3] since start1 = 2 is the smallest start that is >= end2 = 2.

Example 3: Input: intervals = [[1,4],[2,3],[3,4]] Output: [-1,2,-1] Explanation: There is no right interval for [1,4] and [3,4]. The right interval for [2,3] is [3,4] since start2 = 3 is the smallest start that is >= end1 = 3.

Constraints:

1 <= intervals.length <= 2 * 104 intervals[i].length == 2 -106 <= starti <= endi <= 106 The start point of each interval is unique.

==
Problem Number: 736 URL: https://leetcode.com/problems/path-sum-iii Title: 437. Path Sum III Problem Description: Given the root of a binary tree and an integer targetSum, return the number of paths where the sum of the values along the path equals targetSum. The path does not need to start or end at the root or a leaf, but it must go downwards (i.e., traveling only from parent nodes to child nodes).   Example 1:

Input: root = [10,5,-3,3,2,null,11,3,-2,null,1], targetSum = 8 Output: 3 Explanation: The paths that sum to 8 are shown.

Example 2: Input: root = [5,4,8,11,null,13,4,7,2,null,null,5,1], targetSum = 22 Output: 3

Constraints:

The number of nodes in the tree is in the range [0, 1000]. -109 <= Node.val <= 109 -1000 <= targetSum <= 1000

===
Problem Number: 737 URL: https://leetcode.com/problems/find-all-anagrams-in-a-string Title: 438. Find All Anagrams in a String Problem Description: Given two strings s and p, return an array of all the start indices of p's anagrams in s. You may return the answer in any order. An Anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once. Example 1: Input: s = "cbaebabacd", p = "abc" Output: [0,6] Explanation: The substring with start index = 0 is "cba", which is an anagram of "abc". The substring with start index = 6 is "bac", which is an anagram of "abc".

Example 2: Input: s = "abab", p = "ab" Output: [0,1,2] Explanation: The substring with start index = 0 is "ab", which is an anagram of "ab". The substring with start index = 1 is "ba", which is an anagram of "ab". The substring with start index = 2 is "ab", which is an anagram of "ab".

Constraints:

1 <= s.length, p.length <= 3 * 104 s and p consist of lowercase English letters.

===
Problem Number: 738 URL: https://leetcode.com/problems/find-all-duplicates-in-an-array Title: 442. Find All Duplicates in an Array Problem Description: Given an integer array nums of length n where all the integers of nums are in the range [1, n] and each integer appears once or twice, return an array of all the integers that appears twice. You must write an algorithm that runs in O(n) time and uses only constant extra space. Example 1: Input: nums = [4,3,2,7,8,2,3,1] Output: [2,3] Example 2: Input: nums = [1,1,2] Output: [1] Example 3: Input: nums = [1] Output: []

Constraints:

n == nums.length 1 <= n <= 105 1 <= nums[i] <= n Each element in nums appears once or twice.

===
Problem Number: 739 URL: https://leetcode.com/problems/string-compression Title: 443. String Compression Problem Description: Given an array of characters chars, compress it using the following algorithm: Begin with an empty string s. For each group of consecutive repeating characters in chars:

If the group's length is 1, append the character to s. Otherwise, append the character followed by the group's length.

The compressed string s should not be returned separately, but instead, be stored in the input character array chars. Note that group lengths that are

10 or longer will be split into multiple characters in chars. After you are done modifying the input array, return the new length of the array. You must write an algorithm that uses only constant extra space. Example 1: Input: chars = ["a","a","b","b","c","c","c"] Output: Return 6, and the first 6 characters of the input array should be: ["a","2","b","2","c","3"] Explanation: The groups are "aa", "bb", and "ccc". This compresses to "a2b2c3".

Example 2: Input: chars = ["a"] Output: Return 1, and the first character of the input array should be: ["a"] Explanation: The only group is "a", which remains uncompressed since it's a single character.

Example 3: Input: chars = ["a","b","b","b","b","b","b","b","b","b","b","b","b"] Output: Return 4, and the first 4 characters of the input array should be: ["a","b","1","2"]. Explanation: The groups are "a" and "bbbbbbbbbbbb". This compresses to "ab12". Constraints:

1 <= chars.length <= 2000 chars[i] is a lowercase English letter, uppercase English letter, digit, or symbol.

==
Problem Number: 740 URL: https://leetcode.com/problems/add-two-numbers-ii Title: 445. Add Two Numbers II Problem Description: You are given two non-empty linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list. You may assume the two numbers do not contain any leading zero, except the number 0 itself. Example 1:

Input: l1 = [7,2,4,3], l2 = [5,6,4] Output: [7,8,0,7]

Example 2: Input: l1 = [2,4,3], l2 = [5,6,4] Output: [8,0,7]

Example 3: Input: l1 = [0], l2 = [0] Output: [0]

Constraints:

The number of nodes in each linked list is in the range [1, 100]. 0 <= Node.val <= 9 It is guaranteed that the list represents a number that does not have leading zeros.

Follow up: Could you solve it without reversing the input lists?

==
Problem Number: 741 URL: https://leetcode.com/problems/number-of-boomerangs Title: 447. Number of Boomerangs Problem Description: You are given n points in the plane that are all distinct, where points[i] = [xi, yi]. A boomerang is a tuple of points (i, j, k) such that the distance between i and j equals the distance between i and k (the order of the tuple matters). Return the number of boomerangs. Example 1: Input: points = [[0,0],[1,0],[2,0]] Output: 2 Explanation: The two boomerangs are [[1,0],[0,0],[2,0]] and [[1,0],[2,0],[0,0]].

Example 2: Input: points = [[1,1],[2,2],[3,3]] Output: 2

Example 3: Input: points = [[1,1]] Output: 0

Constraints:

n == points.length 1 <= n <= 500 points[i].length == 2 -104 <= xi, yi <= 104 All the points are unique.

==
Problem Number: 742 URL: https://leetcode.com/problems/serialize-and-deserialize-bst Title: 449. Serialize and Deserialize BST Problem Description: Serialization is converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment. Design an algorithm to serialize and deserialize a binary search tree. There is no restriction on how your serialization/deserialization algorithm should work. You need to ensure that a binary search tree can be serialized to a string, and this string can be deserialized to the original tree structure. The encoded string should be as compact as possible. Example 1: Input: root = [2,1,3] Output: [2,1,3] Example 2: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 104]. 0 <= Node.val <= 104 The input tree is guaranteed to be a binary search tree.

==
Problem Number: 743 URL: https://leetcode.com/problems/delete-node-in-a-bst Title: 450. Delete Node in a BST Problem Description: Given a root node reference of a BST and a key, delete the node with the given key in the BST. Return the root node reference (possibly updated) of the BST. Basically, the deletion can be divided into two stages:

Search for a node to remove. If the node is found, delete the node.

Example 1:

Input: root = [5,3,6,2,4,null,7], key = 3 Output: [5,4,6,2,null,null,7] Explanation: Given key to delete is 3. So we find the node with value 3 and delete it. One valid answer is [5,4,6,2,null,null,7], shown in the above BST. Please notice that another valid answer is [5,2,6,null,4,null,7] and it's also accepted.

Example 2: Input: root = [5,3,6,2,4,null,7], key = 0 Output: [5,3,6,2,4,null,7] Explanation: The tree does not contain a node with value = 0.

Example 3: Input: root = [], key = 0 Output: []

Constraints:

The number of nodes in the tree is in the range [0, 104]. -105 <= Node.val <= 105 Each node has a unique value. root is a valid binary search tree. -105 <= key <= 105

Follow up: Could you solve it with time complexity O(height of tree)?

==
Problem Number: 744 URL: https://leetcode.com/problems/sort-characters-by-frequency Title: 451. Sort Characters By Frequency Problem Description: Given a string s, sort it in decreasing order based on the frequency of the characters. The frequency of a character is the number of times it appears in the string. Return the sorted string. If there are multiple answers, return any of them. Example 1: Input: s = "tree" Output: "eert" Explanation: 'e' appears twice while 'r' and 't' both appear once. So 'e' must appear before both 'r' and 't'. Therefore "eetr" is also a valid answer.

Example 2: Input: s = "cccaaa" Output: "aaaccc" Explanation: Both 'c' and 'a' appear three times, so both "cccaaa" and "aaaccc" are valid answers. Note that "cacaca" is incorrect, as the same characters must be together.

Example 3: Input: s = "Aabb" Output: "bbAa" Explanation: "bbaA" is also a valid answer, but "Aabb" is incorrect. Note that 'A' and 'a' are treated as two different characters.

Constraints:

1 <= s.length <= 5 * 105 s consists of uppercase and lowercase English letters and digits.

==
Problem Number: 745 URL: https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons Title: 452. Minimum Number of Arrows to Burst Balloons Problem Description: There are some spherical balloons taped onto a flat wall that represents the XY-plane. The balloons are represented as a 2D integer array points where points[i] = [xstart, xend] denotes a balloon whose horizontal diameter stretches between xstart and xend. You do not know the exact y-coordinates of the balloons. Arrows can be shot up directly vertically (in the positive y-direction) from different points along the x-axis. A balloon with xstart and xend is burst by an arrow shot at x if xstart <= x <= xend. There is no limit to the number of arrows that can be shot. A shot arrow keeps traveling up infinitely, bursting any balloons in its path. Given the array points, return the minimum number of arrows that must be shot to burst all balloons. Example 1: Input: points = [[10,16],[2,8],[1,6],[7,12]] Output: 2 Explanation: The balloons can be burst by 2 arrows: - Shoot an arrow at x = 6, bursting the balloons [2,8] and [1,6]. - Shoot an arrow at x = 11, bursting the balloons [10,16] and [7,12].

Example 2: Input: points = [[1,2],[3,4],[5,6],[7,8]] Output: 4 Explanation: One arrow needs to be shot for each balloon for a total of 4 arrows.

Example 3: Input: points = [[1,2],[2,3],[3,4],[4,5]] Output: 2 Explanation: The balloons can be burst by 2 arrows: - Shoot an arrow at x = 2, bursting the balloons [1,2] and [2,3]. - Shoot an arrow at x = 4, bursting the balloons [3,4] and [4,5].

Constraints:

1 <= points.length <= 105 points[i].length == 2 -231 <= xstart < xend <= 231 - 1

==
Problem Number: 746 URL: https://leetcode.com/problems/minimum-moves-to-equal-array-elements Title: 453. Minimum Moves to Equal Array Elements Problem Description: Given an integer array nums of size n, return the minimum number of moves required to make all array elements equal. In one move, you can increment n - 1 elements of the array by 1. Example 1: Input: nums = [1,2,3] Output: 3 Explanation: Only three moves are needed (remember each move increments two elements): [1,2,3] => [2,3,3] => [3,4,3] => [4,4,4]

Example 2: Input: nums = [1,1,1] Output: 0

Constraints:

n == nums.length 1 <= nums.length <= 105 -109 <= nums[i] <= 109 The answer is guaranteed to fit in a 32-bit integer.

==
Problem Number: 747 URL: https://leetcode.com/problems/4sum-ii Title: 454. 4Sum II Problem Description: Given four integer arrays nums1, nums2, nums3, and nums4 all of length n, return the number of tuples (i, j, k, l) such that:

0 <= i, j, k, l < n nums1[i] + nums2[j] + nums3[k] + nums4[l] == 0

Example 1: Input: nums1 = [1,2], nums2 = [-2,-1], nums3 = [-1,2], nums4 = [0,2] Output: 2 Explanation: The two tuples are: 1. (0, 0, 0, 1) -> nums1[0] + nums2[0] + nums3[0] + nums4[1] = 1 + (-2) + (-1) + 2 = 0 2. (1, 1, 0, 0) -> nums1[1] + nums2[1] + nums3[0] + nums4[0] = 2 + (-1) + (-1) + 0 = 0

Example 2: Input: nums1 = [0], nums2 = [0], nums3 = [0], nums4 = [0] Output: 1

Constraints:

n == nums1.length n == nums2.length n == nums3.length n == nums4.length 1 <= n <= 200 -228 <= nums1[i], nums2[i], nums3[i], nums4[i] <= 228

==
Problem Number: 748 URL: https://leetcode.com/problems/132-pattern Title: 456. 132 Pattern Problem Description: Given an array of n integers nums, a 132 pattern is a subsequence of three integers nums[i], nums[j] and nums[k] such that i < j < k and nums[i] < nums[k] < nums[j]. Return true if there is a 132 pattern in nums, otherwise, return false. Example 1: Input: nums = [1,2,3,4] Output: false Explanation: There is no 132 pattern in the sequence.

Example 2: Input: nums = [3,1,4,2] Output: true Explanation: There is a 132 pattern in the sequence: [1, 4, 2].

Example 3: Input: nums = [-1,3,2,0] Output: true Explanation: There are three 132 patterns in the sequence: [-1, 3, 2], [-1, 3, 0] and [-1, 2, 0].

Constraints:

n == nums.length 1 <= n <= 2 * 105 -109 <= nums[i] <= 109

==

Problem Number: 749 URL: https://leetcode.com/problems/circular-array-loop Title: 457. Circular Array Loop Problem Description: You are playing a game involving a circular array of non-zero integers nums. Each nums[i] denotes the number of indices forward/backward you must move if you are located at index i:

If nums[i] is positive, move nums[i] steps forward, and If nums[i] is negative, move nums[i] steps backward.

Since the array is circular, you may assume that moving forward from the last element puts you on the first element, and moving backwards from the first element puts you on the last element. A cycle in the array consists of a sequence of indices seq of length k where:

Following the movement rules above results in the repeating index sequence seq[0] -> seq[1] -> ... -> seq[k - 1] -> seq[0] -> ... Every nums[seq[j]] is either all positive or all negative. k > 1

Return true if there is a cycle in nums, or false otherwise.   Example 1:

Input: nums = [2,-1,1,2,2] Output: true Explanation: The graph shows how the indices are connected. White nodes are jumping forward, while red is jumping backward. We can see the cycle 0 --> 2 --> 3 --> 0 --> ..., and all of its nodes are white (jumping in the same direction).

Example 2:

Input: nums = [-1,-2,-3,-4,-5,6] Output: false Explanation: The graph shows how the indices are connected. White nodes are jumping forward, while red is jumping backward. The only cycle is of size 1, so we return false.

Example 3:

Input: nums = [1,-1,5,1,4] Output: true Explanation: The graph shows how the indices are connected. White nodes are jumping forward, while red is jumping backward. We can see the cycle 0 --> 1 --> 0 --> ..., and while it is of size > 1, it has a node jumping forward and a node jumping backward, so it is not a cycle. We can see the cycle 3 --> 4 --> 3 --> ..., and all of its nodes are white (jumping in the same direction).

Constraints:

1 <= nums.length <= 5000 -1000 <= nums[i] <= 1000 nums[i] != 0

Follow up: Could you solve it in O(n) time complexity and O(1) extra space complexity?

===
Problem Number: 750 URL: https://leetcode.com/problems/minimum-moves-to-equal-array-elements-ii Title: 462. Minimum Moves to Equal Array Elements II Problem Description: Given an integer array nums of size n, return the minimum number of moves required to make all array elements equal. In one move, you can increment or decrement an element of the array by 1. Test cases are designed so that the answer will fit in a 32-bit integer.   Example 1: Input: nums = [1,2,3] Output: 2 Explanation: Only two moves are needed (remember each move increments or decrements one element): [1,2,3] => [2,2,3] => [2,2,2]

Example 2: Input: nums = [1,10,2,9] Output: 16

Constraints:

n == nums.length 1 <= nums.length <= 105 -109 <= nums[i] <= 109

===
Problem Number: 751 URL: https://leetcode.com/problems/can-i-win Title: 464. Can I Win Problem Description: In the "100 game" two players take turns adding, to a running total, any integer from 1 to 10. The player who first causes the running total to reach or exceed 100 wins. What if we change the game so that players cannot re-use integers? For example, two players might take turns drawing from a common pool of numbers from 1 to 15 without replacement until they reach a total >= 100. Given two integers maxChoosableInteger and desiredTotal, return true if the first player to move can force a win, otherwise, return false. Assume both players play optimally.   Example 1: Input: maxChoosableInteger = 10, desiredTotal = 11 Output: false Explanation: No matter which integer the first player choose, the first player will lose. The first player can choose an integer from 1 up to 10. If the first player choose 1, the second player can only choose integers from 2 up to 10. The second player will win by choosing 10 and get a total = 11, which is >= desiredTotal. Same with other integers chosen by the first player, the second player will always win.

Example 2: Input: maxChoosableInteger = 10, desiredTotal = 0 Output: true

Example 3: Input: maxChoosableInteger = 10, desiredTotal = 1 Output: true

Constraints:

1 <= maxChoosableInteger <= 20 0 <= desiredTotal <= 300

===
Problem Number: 752 URL: https://leetcode.com/problems/unique-substrings-in-wraparound-string Title: 467. Unique Substrings in Wraparound String Problem Description: We define the string base to be the infinite wraparound string of "abcdefghijklmnopqrstuvwxyz", so base will look like this:

"...zabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcd....".

Given a string s, return the number of unique non-empty substrings of s are present in base. Example 1: Input: s = "a" Output: 1 Explanation: Only the substring "a" of s is in base.

Example 2: Input: s = "cac" Output: 2 Explanation: There are two substrings ("a", "c") of s in base.

Example 3: Input: s = "zab" Output: 6 Explanation: There are six substrings ("z", "a", "b", "za", "ab", and "zab") of s in base.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters.

==
Problem Number: 753 URL: https://leetcode.com/problems/validate-ip-address Title: 468. Validate IP Address Problem Description: Given a string queryIP, return "IPv4" if IP is a valid IPv4 address, "IPv6" if IP is a valid IPv6 address or "Neither" if IP is not a correct IP of any type. A valid IPv4 address is an IP in the form "x1.x2.x3.x4" where 0 <= xi <= 255 and xi cannot contain leading zeros. For example, "192.168.1.1" and "192.168.1.0" are valid IPv4 addresses while "192.168.01.1", "192.168.1.00", and "192.168@1.1" are invalid IPv4 addresses. A valid IPv6 address is an IP in the form "x1:x2:x3:x4:x5:x6:x7:x8" where:

1 <= xi.length <= 4 xi is a hexadecimal string which may contain digits, lower-case English letter ('a' to 'f') and upper-case English letters ('A' to 'F'). Leading zeros are allowed in xi.

For example, "2001:0db8:85a3:0000:0000:8a2e:0370:7334" and "2001:db8:85a3:0:0:8A2E:0370:7334" are valid IPv6 addresses, while "2001:0db8:85a3::8A2E:037j:7334" and "02001:0db8:85a3:0000:0000:8a2e:0370:7334" are invalid IPv6 addresses. Example 1: Input: queryIP = "172.16.254.1" Output: "IPv4" Explanation: This is a valid IPv4 address, return "IPv4".

Example 2: Input: queryIP = "2001:0db8:85a3:0:0:8A2E:0370:7334" Output: "IPv6" Explanation: This is a valid IPv6 address, return "IPv6".

Example 3: Input: queryIP = "256.256.256.256" Output: "Neither" Explanation: This is neither a IPv4 address nor a IPv6 address.

Constraints:

queryIP consists only of English letters, digits and the characters '.' and ':'.

==
Problem Number: 754 URL: https://leetcode.com/problems/implement-rand10-using-rand7 Title: 470. Implement Rand10() Using Rand7() Problem Description: Given the API rand7() that generates a uniform random integer in the range [1, 7], write a function rand10() that generates a uniform random integer in the range [1, 10]. You can only call the API rand7(), and you shouldn't call any other API. Please do not use a language's built-in random

API. Each test case will have one internal argument n, the number of times that your implemented function rand10() will be called while testing. Note that this is not an argument passed to rand10(). Example 1: Input: n = 1 Output: [2] Example 2: Input: n = 2 Output: [2,8] Example 3: Input: n = 3 Output: [3,8,10]

Constraints:

1 <= n <= 105

Follow up:

What is the expected value for the number of calls to rand7() function? Could you minimize the number of calls to rand7()?

==
Problem Number: 755 URL: https://leetcode.com/problems/matchsticks-to-square Title: 473. Matchsticks to Square Problem Description: You are given an integer array matchsticks where matchsticks[i] is the length of the ith matchstick. You want to use all the matchsticks to make one square. You should not break any stick, but you can link them up, and each matchstick must be used exactly one time. Return true if you can make this square and false otherwise. Example 1:

Input: matchsticks = [1,1,2,2,2] Output: true Explanation: You can form a square with length 2, one side of the square came two sticks with length 1.

Example 2: Input: matchsticks = [3,3,3,3,4] Output: false Explanation: You cannot find a way to form a square with all the matchsticks.

Constraints:

1 <= matchsticks.length <= 15 1 <= matchsticks[i] <= 108

==
Problem Number: 756 URL: https://leetcode.com/problems/ones-and-zeroes Title: 474. Ones and Zeroes Problem Description: You are given an array of binary strings strs and two integers m and n. Return the size of the largest subset of strs such that there are at most m 0's and n 1's in the subset. A set x is a subset of a set y if all elements of x are also elements of y. Example 1: Input: strs = ["10","0001","111001","1","0"], m = 5, n = 3 Output: 4 Explanation: The largest subset with at most 5 0's and 3 1's is {"10", "0001", "1", "0"}, so the answer is 4. Other valid but smaller subsets include {"0001", "1"} and {"10", "1", "0"}. {"111001"} is an invalid subset because it contains 4 1's, greater than the maximum of 3.

Example 2: Input: strs = ["10","0","1"], m = 1, n = 1 Output: 2 Explanation: The largest subset is {"0", "1"}, so the answer is 2.

Constraints:

1 <= strs.length <= 600 1 <= strs[i].length <= 100 strs[i] consists only of digits '0' and '1'. 1 <= m, n <= 100

==
Problem Number: 757 URL: https://leetcode.com/problems/heaters Title: 475. Heaters Problem Description: Winter is coming! During the contest, your first job is to design a standard heater with a fixed warm radius to warm all the houses. Every house can be warmed, as long as the house is within the heater's warm radius range. Given the positions of houses and heaters on a horizontal line, return the minimum radius standard of heaters so that those heaters could cover all houses. Notice that all the heaters follow your radius standard, and the warm radius will the same. Example 1: Input: houses = [1,2,3], heaters = [2] Output: 1 Explanation: The only heater was placed in the position 2, and if we use the radius 1 standard, then all the houses can be warmed.

Example 2: Input: houses = [1,2,3,4], heaters = [1,4] Output: 1 Explanation: The two heaters were placed at positions 1 and 4. We need to use a radius 1 standard, then all the houses can be warmed.

Example 3: Input: houses = [1,5], heaters = [2] Output: 3

Constraints:

1 <= houses.length, heaters.length <= 3 * 104 1 <= houses[i], heaters[i] <= 109

==
Problem Number: 758 URL: https://leetcode.com/problems/total-hamming-distance Title: 477. Total Hamming Distance Problem Description: The Hamming distance between two integers is the number of positions at which the corresponding bits are different. Given an integer array nums, return the sum of Hamming distances between all the pairs of the integers in nums. Example 1: Input: nums = [4,14,2] Output: 6 Explanation: In binary representation, the 4 is 0100, 14 is 1110, and 2 is 0010 (just showing the four bits relevant in this case). The answer will be: HammingDistance(4, 14) + HammingDistance(4, 2) + HammingDistance(14, 2) = 2 + 2 + 2 = 6.

Example 2: Input: nums = [4,14,4] Output: 4

Constraints:

1 <= nums.length <= 104 0 <= nums[i] <= 109 The answer for the given input will fit in a 32-bit integer.

==
Problem Number: 759 URL: https://leetcode.com/problems/generate-random-point-in-a-circle Title: 478. Generate Random Point in a Circle Problem Description: Given the radius and the position of the center of a circle, implement the function randPoint which generates a uniform random point inside the circle. Implement the Solution class:

Solution(double radius, double x_center, double y_center) initializes the object with the radius of the circle radius and the position of the center (x_center, y_center). randPoint() returns a random point inside the circle. A point on the circumference of the circle is considered to be in the circle. The answer is returned as an array [x, y].

Example 1: Input ["Solution", "randPoint", "randPoint", "randPoint"] [[1.0, 0.0, 0.0], [], [], []] Output [null, [-0.02493, -0.38077], [0.82314, 0.38945], [0.36572, 0.17248]]

Explanation Solution solution = new Solution(1.0, 0.0, 0.0); solution.randPoint(); // return [-0.02493, -0.38077] solution.randPoint(); // return [0.82314, 0.38945] solution.randPoint(); // return [0.36572, 0.17248]

Constraints:

0 < radius <= 108 -107 <= x_center, y_center <= 107 At most 3 * 104 calls will be made to randPoint.

==
Problem Number: 760 URL: https://leetcode.com/problems/magical-string Title: 481. Magical String Problem Description: A magical string s consists of only '1' and '2' and obeys the following rules:

The string s is magical because concatenating the number of contiguous occurrences of characters '1' and '2' generates the string s itself.

The first few elements of s is s = "1221121221221121122……". If we group the consecutive 1's and 2's in s, it will be "1 22 11 2 1 22 1 22 11 2 11 22 ......" and the occurrences of 1's or 2's in each group are "1 2 2 1 1 2 1 2 2 1 2 2 ......". You can see that the occurrence sequence is s itself. Given an integer n, return the number of 1's in the first n number in the magical string s. Example 1: Input: n = 6 Output: 3 Explanation: The first 6 elements of magical string s is "122112" and it contains three 1's, so return 3.

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 105

==
Problem Number: 761 URL: https://leetcode.com/problems/predict-the-winner Title: 486. Predict the Winner Problem Description: You are given an integer array nums. Two players are playing a game with this array: player 1 and player 2. Player 1 and player 2 take turns, with player 1 starting first. Both players start the game with a score of 0. At each turn, the player takes one of the numbers from either end of the array (i.e., nums[0] or nums[nums.length - 1]) which reduces the size of the array by 1. The player adds the chosen number to their score. The game ends when there are no more elements in the array. Return true if Player 1 can win the game. If the scores of both

players are equal, then player 1 is still the winner, and you should also return true. You may assume that both players are playing optimally. Example 1: Input: nums = [1,5,2] Output: false Explanation: Initially, player 1 can choose between 1 and 2. If he chooses 2 (or 1), then player 2 can choose from 1 (or 2) and 5. If player 2 chooses 5, then player 1 will be left with 1 (or 2). So, final score of player 1 is 1 + 2 = 3, and player 2 is 5. Hence, player 1 will never be the winner and you need to return false.

Example 2: Input: nums = [1,5,233,7] Output: true Explanation: Player 1 first chooses 1. Then player 2 has to choose between 5 and 7. No matter which number player 2 choose, player 1 can choose 233. Finally, player 1 has more score (234) than player 2 (12), so you need to return True representing player1 can win.

Constraints:

1 <= nums.length <= 20 0 <= nums[i] <= 107

==
Problem Number: 762 URL: https://leetcode.com/problems/non-decreasing-subsequences Title: 491. Non-decreasing Subsequences Problem Description: Given an integer array nums, return all the different possible non-decreasing subsequences of the given array with at least two elements. You may return the answer in any order. Example 1: Input: nums = [4,6,7,7] Output: [[4,6],[4,6,7],[4,6,7,7],[4,7],[4,7,7],[6,7],[6,7,7],[7,7]]

Example 2: Input: nums = [4,4,3,2,1] Output: [[4,4]]

Constraints:

1 <= nums.length <= 15 -100 <= nums[i] <= 100

==
Problem Number: 763 URL: https://leetcode.com/problems/target-sum Title: 494. Target Sum Problem Description: You are given an integer array nums and an integer target. You want to build an expression out of nums by adding one of the symbols '+' and '-' before each integer in nums and then concatenate all the integers.

For example, if nums = [2, 1], you can add a '+' before 2 and a '-' before 1 and concatenate them to build the expression "+2-1".

Return the number of different expressions that you can build, which evaluates to target. Example 1: Input: nums = [1,1,1,1,1], target = 3 Output: 5 Explanation: There are 5 ways to assign symbols to make the sum of nums be target 3. -1 + 1 + 1 + 1 + 1 = 3 +1 - 1 + 1 + 1 + 1 = 3 +1 + 1 - 1 + 1 + 1 = 3 +1 + 1 + 1 - 1 + 1 = 3 +1 + 1 + 1 + 1 - 1 = 3

Example 2: Input: nums = [1], target = 1 Output: 1

Constraints:

1 <= nums.length <= 20 0 <= nums[i] <= 1000 0 <= sum(nums[i]) <= 1000 -1000 <= target <= 1000

==

Problem Number: 764 URL: https://leetcode.com/problems/random-point-in-non-overlapping-rectangles Title: 497. Random Point in Non-overlapping Rectangles Problem Description: You are given an array of non-overlapping axis-aligned rectangles rects where rects[i] = [ai, bi, xi, yi] indicates that (ai, bi) is the bottom-left corner point of the ith rectangle and (xi, yi) is the top-right corner point of the ith rectangle. Design an algorithm to pick a random integer point inside the space covered by one of the given rectangles. A point on the perimeter of a rectangle is included in the space covered by the rectangle. Any integer point inside the space covered by one of the given rectangles should be equally likely to be returned. Note that an integer point is a point that has integer coordinates. Implement the Solution class:

Solution(int[][] rects) Initializes the object with the given rectangles rects. int[] pick() Returns a random integer point [u, v] inside the space covered by one of the given rectangles.

Example 1:

Input ["Solution", "pick", "pick", "pick", "pick", "pick"] [[[[-2, -2, 1, 1], [2, 2, 4, 6]]], [], [], [], [], []] Output [null, [1, -2], [1, -1], [-1, -2], [-2, -2], [0, 0]]

Explanation Solution solution = new Solution([[-2, -2, 1, 1], [2, 2, 4, 6]]); solution.pick(); // return [1, -2] solution.pick(); // return [1, -1] solution.pick(); // return [-1, -2] solution.pick(); // return [-2, -2] solution.pick(); // return [0, 0]

Constraints:

1 <= rects.length <= 100 rects[i].length == 4 -109 <= ai < xi <= 109 -109 <= bi < yi <= 109 xi - ai <= 2000 yi - bi <= 2000 All the rectangles do not overlap. At most 104 calls will be made to pick.

==

Problem Number: 765 URL: https://leetcode.com/problems/diagonal-traverse Title: 498. Diagonal Traverse Problem Description: Given an m x n matrix mat, return an array of all the elements of the array in a diagonal order. Example 1:

Input: mat = [[1,2,3],[4,5,6],[7,8,9]] Output: [1,2,4,7,5,3,6,8,9]

Example 2: Input: mat = [[1,2],[3,4]] Output: [1,2,3,4]

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 104 1 <= m * n <= 104 -105 <= mat[i][j] <= 105

==

Problem Number: 766 URL: https://leetcode.com/problems/next-greater-element-ii Title: 503. Next Greater Element II Problem Description: Given a circular integer array nums (i.e., the next element of nums[nums.length - 1] is nums[0]), return the next greater number for every element in nums. The next greater number of a number x is the first greater number to its traversing-order next in the array, which means you could search circularly to find its next greater number. If it doesn't exist, return -1 for this number. Example 1: Input: nums = [1,2,1] Output: [2,-1,2] Explanation: The first 1's next greater number is 2; The number 2 can't find next greater number. The second 1's next greater number needs to search circularly, which is also 2.

Example 2: Input: nums = [1,2,3,4,3] Output: [2,3,4,-1,4]

Constraints:

1 <= nums.length <= 104 -109 <= nums[i] <= 109

==
Problem Number: 767 URL: https://leetcode.com/problems/most-frequent-subtree-sum Title: 508. Most Frequent Subtree Sum Problem Description: Given the root of a binary tree, return the most frequent subtree sum. If there is a tie, return all the values with the highest frequency in any order. The subtree sum of a node is defined as the sum of all the node values formed by the subtree rooted at that node (including the node itself). Example 1:

Input: root = [5,2,-3] Output: [2,-3,4]

Example 2:

Input: root = [5,2,-5] Output: [2]

Constraints:

The number of nodes in the tree is in the range [1, 104]. -105 <= Node.val <= 105

==
Problem Number: 768 URL: https://leetcode.com/problems/find-bottom-left-tree-value Title: 513. Find Bottom Left Tree Value Problem Description: Given the root of a binary tree, return the leftmost value in the last row of the tree. Example 1:

Input: root = [2,1,3] Output: 1

Example 2:

Input: root = [1,2,3,4,null,5,6,null,null,7] Output: 7

Constraints:

The number of nodes in the tree is in the range [1, 104]. -231 <= Node.val <= 231 - 1

==
Problem Number: 769 URL: https://leetcode.com/problems/find-largest-value-

in-each-tree-row Title: 515. Find Largest Value in Each Tree Row Problem Description: Given the root of a binary tree, return an array of the largest value in each row of the tree (0-indexed).   Example 1:

Input: root = [1,3,2,5,3,null,9] Output: [1,3,9]

Example 2: Input: root = [1,2,3] Output: [1,3]

Constraints:

The number of nodes in the tree will be in the range [0, 104]. -231 <= Node.val <= 231 - 1

==
Problem Number: 770 URL: https://leetcode.com/problems/longest-palindromic-subsequence Title: 516. Longest Palindromic Subsequence Problem Description: Given a string s, find the longest palindromic subsequence's length in s. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.   Example 1: Input: s = "bbbab" Output: 4 Explanation: One possible longest palindromic subsequence is "bbbb".

Example 2: Input: s = "cbbd" Output: 2 Explanation: One possible longest palindromic subsequence is "bb".

Constraints:

1 <= s.length <= 1000 s consists only of lowercase English letters.

==
Problem Number: 771 URL: https://leetcode.com/problems/coin-change-ii Title: 518. Coin Change II Problem Description: You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money. Return the number of combinations that make up that amount. If that amount of money cannot be made up by any combination of the coins, return 0. You may assume that you have an infinite number of each kind of coin. The answer is guaranteed to fit into a signed 32-bit integer.   Example 1: Input: amount = 5, coins = [1,2,5] Output: 4 Explanation: there are four ways to make up the amount: 5=5 5=2+2+1 5=2+1+1+1 5=1+1+1+1+1

Example 2: Input: amount = 3, coins = [2] Output: 0 Explanation: the amount of 3 cannot be made up just with coins of 2.

Example 3: Input: amount = 10, coins = [10] Output: 1

Constraints:

1 <= coins.length <= 300 1 <= coins[i] <= 5000 All the values of coins are unique. 0 <= amount <= 5000

==
Problem Number: 772 URL: https://leetcode.com/problems/random-flip-

matrix Title: 519. Random Flip Matrix Problem Description: There is an m x n binary grid matrix with all the values set 0 initially. Design an algorithm to randomly pick an index (i, j) where matrix[i][j] == 0 and flips it to 1. All the indices (i, j) where matrix[i][j] == 0 should be equally likely to be returned. Optimize your algorithm to minimize the number of calls made to the built-in random function of your language and optimize the time and space complexity. Implement the Solution class:

Solution(int m, int n) Initializes the object with the size of the binary matrix m and n. int[] flip() Returns a random index [i, j] of the matrix where matrix[i][j] == 0 and flips it to 1. void reset() Resets all the values of the matrix to be 0.

Example 1: Input ["Solution", "flip", "flip", "flip", "reset", "flip"] [[3, 1], [], [], [], [], []] Output [null, [1, 0], [2, 0], [0, 0], null, [2, 0]]

Explanation Solution solution = new Solution(3, 1); solution.flip(); // return [1, 0], [0,0], [1,0], and [2,0] should be equally likely to be returned. solution.flip(); // return [2, 0], Since [1,0] was returned, [2,0] and [0,0] solution.flip(); // return [0, 0], Based on the previously returned indices, only [0,0] can be returned. solution.reset(); // All the values are reset to 0 and can be returned. solution.flip(); // return [2, 0], [0,0], [1,0], and [2,0] should be equally likely to be returned.

Constraints:

1 <= m, n <= 104 There will be at least one free cell for each call to flip. At most 1000 calls will be made to flip and reset.

==

Problem Number: 773 URL: https://leetcode.com/problems/longest-uncommon-subsequence-ii Title: 522. Longest Uncommon Subsequence II Problem Description: Given an array of strings strs, return the length of the longest uncommon subsequence between them. If the longest uncommon subsequence does not exist, return -1. An uncommon subsequence between an array of strings is a string that is a subsequence of one string but not the others. A subsequence of a string s is a string that can be obtained after deleting any number of characters from s.

For example, "abc" is a subsequence of "aebdc" because you can delete the underlined characters in "aebdc" to get "abc". Other subsequences of "aebdc" include "aebdc", "aeb", and "" (empty string).

Example 1: Input: strs = ["aba","cdc","eae"] Output: 3 Example 2: Input: strs = ["aaa","aaa","aa"] Output: -1

Constraints:

2 <= strs.length <= 50 1 <= strs[i].length <= 10 strs[i] consists of lowercase English letters.

==

Problem Number: 774 URL: https://leetcode.com/problems/continuous-subarray-sum Title: 523. Continuous Subarray Sum Problem Description: Given an integer array nums and an integer k, return true if nums has a good subarray or false otherwise. A good subarray is a subarray where:

its length is at least two, and the sum of the elements of the subarray is a multiple of k.

Note that:

A subarray is a contiguous part of the array. An integer x is a multiple of k if there exists an integer n such that x = n * k. 0 is always a multiple of k.

Example 1: Input: nums = [23,2,4,6,7], k = 6 Output: true Explanation: [2, 4] is a continuous subarray of size 2 whose elements sum up to 6.

Example 2: Input: nums = [23,2,6,4,7], k = 6 Output: true Explanation: [23, 2, 6, 4, 7] is an continuous subarray of size 5 whose elements sum up to 42. 42 is a multiple of 6 because 42 = 7 * 6 and 7 is an integer.

Example 3: Input: nums = [23,2,6,4,7], k = 13 Output: false

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109 0 <= sum(nums[i]) <= 231 - 1 1 <= k <= 231 - 1

==
Problem Number: 775 URL: https://leetcode.com/problems/longest-word-in-dictionary-through-deleting Title: 524. Longest Word in Dictionary through Deleting Problem Description: Given a string s and a string array dictionary, return the longest string in the dictionary that can be formed by deleting some of the given string characters. If there is more than one possible result, return the longest word with the smallest lexicographical order. If there is no possible result, return the empty string. Example 1: Input: s = "abpcplea", dictionary = ["ale","apple","monkey","plea"] Output: "apple"

Example 2: Input: s = "abpcplea", dictionary = ["a","b","c"] Output: "a"

Constraints:

1 <= s.length <= 1000 1 <= dictionary.length <= 1000 1 <= dictionary[i].length <= 1000 s and dictionary[i] consist of lowercase English letters.

==
Problem Number: 776 URL: https://leetcode.com/problems/contiguous-array Title: 525. Contiguous Array Problem Description: Given a binary array nums, return the maximum length of a contiguous subarray with an equal number of 0 and 1. Example 1: Input: nums = [0,1] Output: 2 Explanation: [0, 1] is the longest contiguous subarray with an equal number of 0 and 1.

Example 2: Input: nums = [0,1,0] Output: 2 Explanation: [0, 1] (or [1, 0]) is a longest contiguous subarray with equal number of 0 and 1.

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1.

== Problem Number: 777 URL: https://leetcode.com/problems/beautiful-arrangement Title: 526. Beautiful Arrangement Problem Description: Suppose you have n integers labeled 1 through n. A permutation of those n integers perm (1-indexed) is considered a beautiful arrangement if for every i (1 <= i <= n), either of the following is true:

perm[i] is divisible by i. i is divisible by perm[i].

Given an integer n, return the number of the beautiful arrangements that you can construct. Example 1: Input: n = 2 Output: 2 Explanation: The first beautiful arrangement is [1,2]: - perm[1] = 1 is divisible by i = 1 - perm[2] = 2 is divisible by i = 2 The second beautiful arrangement is [2,1]: - perm[1] = 2 is divisible by i = 1 - i = 2 is divisible by perm[2] = 1

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 15

== Problem Number: 778 URL: https://leetcode.com/problems/random-pick-with-weight Title: 528. Random Pick with Weight Problem Description: You are given a 0-indexed array of positive integers w where w[i] describes the weight of the ith index. You need to implement the function pickIndex(), which randomly picks an index in the range [0, w.length - 1] (inclusive) and returns it. The probability of picking an index i is w[i] / sum(w).

For example, if w = [1, 3], the probability of picking index 0 is 1 / (1 + 3) = 0.25 (i.e., 25%), and the probability of picking index 1 is 3 / (1 + 3) = 0.75 (i.e., 75%).

Example 1: Input ["Solution","pickIndex"] [[[1]],[]] Output [null,0]

Explanation Solution solution = new Solution([1]); solution.pickIndex(); // return 0. The only option is to return 0 since there is only one element in w.

Example 2: Input ["Solution","pickIndex","pickIndex","pickIndex","pickIndex","pickIndex"] [[[1,3]],[],[],[],[],[]] Output [null,1,1,1,1,0]

Explanation Solution solution = new Solution([1, 3]); solution.pickIndex(); // return 1. It is returning the second element (index = 1) that has a probability of 3/4. solution.pickIndex(); // return 1 solution.pickIndex(); // return 1 solution.pickIndex(); // return 1 solution.pickIndex(); // return 0. It is returning the first element (index = 0) that has a probability of 1/4.

Since this is a randomization problem, multiple answers are allowed. All of the following outputs can be considered correct: [null,1,1,1,1,0] [null,1,1,1,1,1] [null,1,1,1,0,0] [null,1,1,1,0,1] [null,1,0,1,0,0] ...... and so on.

Constraints:

1 <= w.length <= 104 1 <= w[i] <= 105 pickIndex will be called at most 104 times.

==
Problem Number: 779 URL: https://leetcode.com/problems/minesweeper Title: 529. Minesweeper Problem Description: Let’s play the minesweeper game (Wikipedia, online game)! You are given an m x n char matrix board representing the game board where:

’M’ represents an unrevealed mine, ’E’ represents an unrevealed empty square, ’B’ represents a revealed blank square that has no adjacent mines (i.e., above, below, left, right, and all 4 diagonals), digit (’1’ to ’8’) represents how many mines are adjacent to this revealed square, and ’X’ represents a revealed mine.

You are also given an integer array click where click = [clickr, clickc] represents the next click position among all the unrevealed squares (’M’ or ’E’). Return the board after revealing this position according to the following rules:

If a mine ’M’ is revealed, then the game is over. You should change it to ’X’. If an empty square ’E’ with no adjacent mines is revealed, then change it to a revealed blank ’B’ and all of its adjacent unrevealed squares should be revealed recursively. If an empty square ’E’ with at least one adjacent mine is revealed, then change it to a digit (’1’ to ’8’) representing the number of adjacent mines. Return the board when no more squares will be revealed.

Example 1:

Input: board = [[”E”,”E”,”E”,”E”,”E”],[”E”,”E”,”M”,”E”,”E”],[”E”,”E”,”E”,”E”,”E”],[”E”,”E”,”E”,”E”,”E”]], click = [3,0] Output: [[”B”,”1”,”E”,”1”,”B”],[”B”,”1”,”M”,”1”,”B”],[”B”,”1”,”1”,”1”,”B”],[”B”,”B”,”B”,”B”,”B

Example 2:

Input: board = [[”B”,”1”,”E”,”1”,”B”],[”B”,”1”,”M”,”1”,”B”],[”B”,”1”,”1”,”1”,”B”],[”B”,”B”,”B”,”B”,”B”]], click = [1,2] Output: [[”B”,”1”,”E”,”1”,”B”],[”B”,”1”,”X”,”1”,”B”],[”B”,”1”,”1”,”1”,”B”],[”B”,”B”,”B”,”B”,”B”

Constraints:

m == board.length n == board[i].length 1 <= m, n <= 50 board[i][j] is either ’M’, ’E’, ’B’, or a digit from ’1’ to ’8’. click.length == 2 0 <= clickr < m 0 <= clickc < n board[clickr][clickc] is either ’M’ or ’E’.

==
Problem Number: 780 URL: https://leetcode.com/problems/k-diff-pairs-in-an-array Title: 532. K-diff Pairs in an Array Problem Description: Given an array of integers nums and an integer k, return the number of unique k-diff pairs in the array. A k-diff pair is an integer pair (nums[i], nums[j]), where the following are true:

0 <= i, j < nums.length i != j |nums[i] - nums[j]| == k

Notice that |val| denotes the absolute value of val. Example 1: Input: nums = [3,1,4,1,5], k = 2 Output: 2 Explanation: There are two 2-diff pairs in the array, (1, 3) and (3, 5). Although we have two 1s in the input, we should only return the number of unique pairs.

Example 2: Input: nums = [1,2,3,4,5], k = 1 Output: 4 Explanation: There are four 1-diff pairs in the array, (1, 2), (2, 3), (3, 4) and (4, 5).

Example 3: Input: nums = [1,3,1,5,4], k = 0 Output: 1 Explanation: There is one 0-diff pair in the array, (1, 1).

Constraints:

1 <= nums.length <= 104 -107 <= nums[i] <= 107 0 <= k <= 107

==

Problem Number: 781 URL: https://leetcode.com/problems/encode-and-decode-tinyurl Title: 535. Encode and Decode TinyURL Problem Description: Note: This is a companion problem to the System Design problem: Design TinyURL. TinyURL is a URL shortening service where you enter a URL such as https://leetcode.com/problems/design-tinyurl and it returns a short URL such as http://tinyurl.com/4e9iAk. Design a class to encode a URL and decode a tiny URL. There is no restriction on how your encode/decode algorithm should work. You just need to ensure that a URL can be encoded to a tiny URL and the tiny URL can be decoded to the original URL. Implement the Solution class:

Solution() Initializes the object of the system. String encode(String longUrl) Returns a tiny URL for the given longUrl. String decode(String shortUrl) Returns the original long URL for the given shortUrl. It is guaranteed that the given shortUrl was encoded by the same object.

Example 1: Input: url = "https://leetcode.com/problems/design-tinyurl" Output: "https://leetcode.com/problems/design-tinyurl"

Explanation: Solution obj = new Solution(); string tiny = obj.encode(url); // returns the encoded tiny url. string ans = obj.decode(tiny); // returns the original url after decoding it.

Constraints:

1 <= url.length <= 104 url is guranteed to be a valid URL.

==

Problem Number: 782 URL: https://leetcode.com/problems/complex-number-multiplication Title: 537. Complex Number Multiplication Problem Description: A complex number can be represented as a string on the form "real+imaginaryi" where:

real is the real part and is an integer in the range [-100, 100]. imaginary is the imaginary part and is an integer in the range [-100, 100]. i2 == -1.

Given two complex numbers num1 and num2 as strings, return a string of the complex number that represents their multiplications. Example 1: Input: num1 = ”1+1i”, num2 = ”1+1i” Output: ”0+2i” Explanation: (1 + i) * (1 + i) = 1 + i2 + 2 * i = 2i, and you need convert it to the form of 0+2i.

Example 2: Input: num1 = ”1+-1i”, num2 = ”1+-1i” Output: ”0+-2i” Explanation: (1 - i) * (1 - i) = 1 + i2 - 2 * i = -2i, and you need convert it to the form of 0+-2i.

Constraints:

num1 and num2 are valid complex numbers.

==

Problem Number: 783 URL: https://leetcode.com/problems/convert-bst-to-greater-tree Title: 538. Convert BST to Greater Tree Problem Description: Given the root of a Binary Search Tree (BST), convert it to a Greater Tree such that every key of the original BST is changed to the original key plus the sum of all keys greater than the original key in BST. As a reminder, a binary search tree is a tree that satisfies these constraints:

The left subtree of a node contains only nodes with keys less than the node’s key. The right subtree of a node contains only nodes with keys greater than the node’s key. Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [4,1,6,0,2,5,7,null,null,null,3,null,null,null,8] Output: [30,36,21,36,35,26,15,null,null,null,33,null,nu

Example 2: Input: root = [0,null,1] Output: [1,null,1]

Constraints:

The number of nodes in the tree is in the range [0, 104]. -104 <= Node.val <= 104 All the values in the tree are unique. root is guaranteed to be a valid binary search tree.

Note: This question is the same as 1038: https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree/

==

Problem Number: 784 URL: https://leetcode.com/problems/minimum-time-difference Title: 539. Minimum Time Difference Problem Description: Given a list of 24-hour clock time points in ”HH:MM” format, return the minimum minutes difference between any two time-points in the list. Example 1: Input: timePoints = [”23:59”,”00:00”] Output: 1 Example 2: Input: timePoints = [”00:00”,”23:59”,”00:00”] Output: 0

Constraints:

2 <= timePoints.length <= 2 * 104 timePoints[i] is in the format ”HH:MM”.

==
Problem Number: 785 URL: https://leetcode.com/problems/single-element-in-a-sorted-array Title: 540. Single Element in a Sorted Array Problem Description: You are given a sorted array consisting of only integers where every element appears exactly twice, except for one element which appears exactly once. Return the single element that appears only once. Your solution must run in O(log n) time and O(1) space. Example 1: Input: nums = [1,1,2,3,3,4,4,8,8] Output: 2 Example 2: Input: nums = [3,3,7,7,10,11,11] Output: 10

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 105

==
Problem Number: 786 URL: https://leetcode.com/problems/01-matrix Title: 542. 01 Matrix Problem Description: Given an m x n binary matrix mat, return the distance of the nearest 0 for each cell. The distance between two adjacent cells is 1. Example 1:

Input: mat = [[0,0,0],[0,1,0],[0,0,0]] Output: [[0,0,0],[0,1,0],[0,0,0]]

Example 2:

Input: mat = [[0,0,0],[0,1,0],[1,1,1]] Output: [[0,0,0],[0,1,0],[1,2,1]]

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 104 1 <= m * n <= 104 mat[i][j] is either 0 or 1. There is at least one 0 in mat.

==
Problem Number: 787 URL: https://leetcode.com/problems/number-of-provinces Title: 547. Number of Provinces Problem Description: There are n cities. Some of them are connected, while some are not. If city a is connected directly with city b, and city b is connected directly with city c, then city a is connected indirectly with city c. A province is a group of directly or indirectly connected cities and no other cities outside of the group. You are given an n x n matrix isConnected where isConnected[i][j] = 1 if the ith city and the jth city are directly connected, and isConnected[i][j] = 0 otherwise. Return the total number of provinces. Example 1:

Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]] Output: 2

Example 2:

Input: isConnected = [[1,0,0],[0,1,0],[0,0,1]] Output: 3

Constraints:

1 <= n <= 200 n == isConnected.length n == isConnected[i].length isConnected[i][j] is 1 or 0. isConnected[i][i] == 1 isConnected[i][j] == isConnected[j][i]

==
Problem Number: 788 URL: https://leetcode.com/problems/optimal-division Title: 553. Optimal Division Problem Description: You are given an integer array nums. The adjacent integers in nums will perform the float division.

For example, for nums = [2,3,4], we will evaluate the expression "2/3/4".

However, you can add any number of parenthesis at any position to change the priority of operations. You want to add these parentheses such the value of the expression after the evaluation is maximum. Return the corresponding expression that has the maximum value in string format. Note: your expression should not contain redundant parenthesis. Example 1: Input: nums = [1000,100,10,2] Output: "1000/(100/10/2)" Explanation: 1000/(100/10/2) = 1000/((100/10)/2) = 200 However, the bold parenthesis in "1000/((100/10)/2)" are redundant since they do not influence the operation priority. So you should return "1000/(100/10/2)". Other cases: 1000/(100/10)/2 = 50 1000/(100/(10/2)) = 50 1000/100/10/2 = 0.5 1000/100/(10/2) = 2

Example 2: Input: nums = [2,3,4] Output: "2/(3/4)" Explanation: (2/(3/4)) = 8/3 = 2.667 It can be shown that after trying all possibilities, we cannot get an expression with evaluation greater than 2.667

Constraints:

1 <= nums.length <= 10 2 <= nums[i] <= 1000 There is only one optimal division for the given input.

==
Problem Number: 789 URL: https://leetcode.com/problems/brick-wall Title: 554. Brick Wall Problem Description: There is a rectangular brick wall in front of you with n rows of bricks. The ith row has some number of bricks each of the same height (i.e., one unit) but they can be of different widths. The total width of each row is the same. Draw a vertical line from the top to the bottom and cross the least bricks. If your line goes through the edge of a brick, then the brick is not considered as crossed. You cannot draw a line just along one of the two vertical edges of the wall, in which case the line will obviously cross no bricks. Given the 2D array wall that contains the information about the wall, return the minimum number of crossed bricks after drawing such a vertical line. Example 1:

Input: wall = [[1,2,2,1],[3,1,2],[1,3,2],[2,4],[3,1,2],[1,3,1,1]] Output: 2

Example 2: Input: wall = [[1],[1],[1]] Output: 3

Constraints:

n == wall.length 1 <= n <= 104 1 <= wall[i].length <= 104 1 <= sum(wall[i].length) <= 2 * 104 sum(wall[i]) is the same for each row i. 1 <= wall[i][j] <= 231 - 1

==

Problem Number: 790 URL: https://leetcode.com/problems/next-greater-element-iii Title: 556. Next Greater Element III Problem Description: Given a positive integer n, find the smallest integer which has exactly the same digits existing in the integer n and is greater in value than n. If no such positive integer exists, return -1. Note that the returned integer should fit in 32-bit integer, if there is a valid answer but it does not fit in 32-bit integer, return -1. Example 1: Input: n = 12 Output: 21 Example 2: Input: n = 21 Output: -1

Constraints:

1 <= n <= 231 - 1

==

Problem Number: 791 URL: https://leetcode.com/problems/logical-or-of-two-binary-grids-represented-as-quad-trees Title: 558. Logical OR of Two Binary Grids Represented as Quad-Trees Problem Description: A Binary Matrix is a matrix in which all the elements are either 0 or 1. Given quadTree1 and quadTree2. quadTree1 represents a n * n binary matrix and quadTree2 represents another n * n binary matrix. Return a Quad-Tree representing the n * n binary matrix which is the result of logical bitwise OR of the two binary matrixes represented by quadTree1 and quadTree2. Notice that you can assign the value of a node to True or False when isLeaf is False, and both are accepted in the answer. A Quad-Tree is a tree data structure in which each internal node has exactly four children. Besides, each node has two attributes:

val: True if the node represents a grid of 1's or False if the node represents a grid of 0's. isLeaf: True if the node is leaf node on the tree or False if the node has the four children.

class Node { public boolean val; public boolean isLeaf; public Node topLeft; public Node topRight; public Node bottomLeft; public Node bottomRight; } We can construct a Quad-Tree from a two-dimensional area using the following steps:

If the current grid has the same value (i.e all 1's or all 0's) set isLeaf True and set val to the value of the grid and set the four children to Null and stop. If the current grid has different values, set isLeaf to False and set val to any value and divide the current grid into four sub-grids as shown in the photo. Recurse for each of the children with the proper sub-grid.

If you want to know more about the Quad-Tree, you can refer to the wiki. Quad-Tree format: The input/output represents the serialized format of a Quad-Tree using level order traversal, where null signifies a path terminator where no node exists below. It is very similar to the serialization of the binary tree. The only difference is that the node is represented as a list [isLeaf, val]. If the value of isLeaf or val is True we represent it as 1 in the list [isLeaf, val] and if the value of isLeaf or val is False we represent it as 0. Example 1:

Input: quadTree1 = [[0,1],[1,1],[1,1],[1,0],[1,0]] , quadTree2 = [[0,1],[1,1],[0,1],[1,1],[1,0],null,null,null,null,[1,0],[1,
Output: [[0,0],[1,1],[1,1],[1,1],[1,0]] Explanation: quadTree1 and quadTree2 are

shown above. You can see the binary matrix which is represented by each Quad-Tree. If we apply logical bitwise OR on the two binary matrices we get the binary matrix below which is represented by the result Quad-Tree. Notice that the binary matrices shown are only for illustration, you don't have to construct the binary matrix to get the result tree.

Example 2: Input: quadTree1 = [[1,0]], quadTree2 = [[1,0]] Output: [[1,0]] Explanation: Each tree represents a binary matrix of size 1*1. Each matrix contains only zero. The resulting matrix is of size 1*1 with also zero.

Constraints:

quadTree1 and quadTree2 are both valid Quad-Trees each representing a n * n grid. n == 2x where 0 <= x <= 9.

==
Problem Number: 792 URL: https://leetcode.com/problems/subarray-sum-equals-k Title: 560. Subarray Sum Equals K Problem Description: Given an array of integers nums and an integer k, return the total number of subarrays whose sum equals to k. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,1,1], k = 2 Output: 2 Example 2: Input: nums = [1,2,3], k = 3 Output: 2

Constraints:

1 <= nums.length <= 2 * 104 -1000 <= nums[i] <= 1000 -107 <= k <= 107

==
Problem Number: 793 URL: https://leetcode.com/problems/array-nesting Title: 565. Array Nesting Problem Description: You are given an integer array nums of length n where nums is a permutation of the numbers in the range [0, n - 1]. You should build a set s[k] = {nums[k], nums[nums[k]], nums[nums[nums[k]]], ... } subjected to the following rule:

The first element in s[k] starts with the selection of the element nums[k] of index = k. The next element in s[k] should be nums[nums[k]], and then nums[nums[nums[k]]], and so on. We stop adding right before a duplicate element occurs in s[k].

Return the longest length of a set s[k]. Example 1: Input: nums = [5,4,0,3,1,6,2] Output: 4 Explanation: nums[0] = 5, nums[1] = 4, nums[2] = 0, nums[3] = 3, nums[4] = 1, nums[5] = 6, nums[6] = 2. One of the longest sets s[k]: s[0] = {nums[0], nums[5], nums[6], nums[2]} = {5, 6, 2, 0}

Example 2: Input: nums = [0,1,2] Output: 1

Constraints:

1 <= nums.length <= 105 0 <= nums[i] < nums.length All the values of nums are unique.

==

Problem Number: 794 URL: https://leetcode.com/problems/permutation-in-string Title: 567. Permutation in String Problem Description: Given two strings s1 and s2, return true if s2 contains a permutation of s1, or false otherwise. In other words, return true if one of s1's permutations is the substring of s2. Example 1: Input: s1 = "ab", s2 = "eidbaooo" Output: true Explanation: s2 contains one permutation of s1 ("ba").

Example 2: Input: s1 = "ab", s2 = "eidboaoo" Output: false

Constraints:

1 <= s1.length, s2.length <= 104 s1 and s2 consist of lowercase English letters.

==

Problem Number: 795 URL: https://leetcode.com/problems/out-of-boundary-paths Title: 576. Out of Boundary Paths Problem Description: There is an m x n grid with a ball. The ball is initially at the position [startRow, startColumn]. You are allowed to move the ball to one of the four adjacent cells in the grid (possibly out of the grid crossing the grid boundary). You can apply at most maxMove moves to the ball. Given the five integers m, n, maxMove, startRow, startColumn, return the number of paths to move the ball out of the grid boundary. Since the answer can be very large, return it modulo 109 + 7. Example 1:

Input: m = 2, n = 2, maxMove = 2, startRow = 0, startColumn = 0 Output: 6

Example 2:

Input: m = 1, n = 3, maxMove = 3, startRow = 0, startColumn = 1 Output: 12

Constraints:

1 <= m, n <= 50 0 <= maxMove <= 50 0 <= startRow < m 0 <= startColumn < n

==

Problem Number: 796 URL: https://leetcode.com/problems/shortest-unsorted-continuous-subarray Title: 581. Shortest Unsorted Continuous Subarray Problem Description: Given an integer array nums, you need to find one continuous subarray such that if you only sort this subarray in non-decreasing order, then the whole array will be sorted in non-decreasing order. Return the shortest such subarray and output its length. Example 1: Input: nums = [2,6,4,8,10,9,15] Output: 5 Explanation: You need to sort [6, 4, 8, 10, 9] in ascending order to make the whole array sorted in ascending order.

Example 2: Input: nums = [1,2,3,4] Output: 0

Example 3: Input: nums = [1] Output: 0

Constraints:

1 <= nums.length <= 104 -105 <= nums[i] <= 105

Follow up: Can you solve it in O(n) time complexity? ==================================== Problem Number: 797 URL: https://leetcode.com/problems/delete-operation-for-two-strings Title: 583. Delete Operation for Two Strings Problem Description: Given two strings word1 and word2, return the minimum number of steps required to make word1 and word2 the same. In one step, you can delete exactly one character in either string. Example 1: Input: word1 = "sea", word2 = "eat" Output: 2 Explanation: You need one step to make "sea" to "ea" and another step to make "eat" to "ea".

Example 2: Input: word1 = "leetcode", word2 = "etco" Output: 4

Constraints:

1 <= word1.length, word2.length <= 500 word1 and word2 consist of only lowercase English letters.

== Problem Number: 798 URL: https://leetcode.com/problems/fraction-addition-and-subtraction Title: 592. Fraction Addition and Subtraction Problem Description: Given a string expression representing an expression of fraction addition and subtraction, return the calculation result in string format. The final result should be an irreducible fraction. If your final result is an integer, change it to the format of a fraction that has a denominator 1. So in this case, 2 should be converted to 2/1. Example 1: Input: expression = "-1/2+1/2" Output: "0/1"

Example 2: Input: expression = "-1/2+1/2+1/3" Output: "1/3"

Example 3: Input: expression = "1/3-1/2" Output: "-1/6"

Constraints:

The input string only contains '0' to '9', '/', '+' and '-'. So does the output. Each fraction (input and output) has the format ±numerator/denominator. If the first input fraction or the output is positive, then '+' will be omitted. The input only contains valid irreducible fractions, where the numerator and denominator of each fraction will always be in the range [1, 10]. If the denominator is 1, it means this fraction is actually an integer in a fraction format defined above. The number of given fractions will be in the range [1, 10]. The numerator and denominator of the final result are guaranteed to be valid and in the range of 32-bit int.

== Problem Number: 799 URL: https://leetcode.com/problems/valid-square Title: 593. Valid Square Problem Description: Given the coordinates of four points in 2D space p1, p2, p3 and p4, return true if the four points construct a square. The coordinate of a point pi is represented as [xi, yi]. The input is not given in any order. A valid square has four equal sides with positive length and four equal angles (90-degree angles).   Example 1: Input: p1 = [0,0], p2 = [1,1], p3 = [1,0], p4 = [0,1] Output: true

Example 2: Input: p1 = [0,0], p2 = [1,1], p3 = [1,0], p4 = [0,12] Output: false

Example 3: Input: p1 = [1,0], p2 = [-1,0], p3 = [0,1], p4 = [0,-1] Output: true

Constraints:

p1.length == p2.length == p3.length == p4.length == 2 -104 <= xi, yi <= 104

==

Problem Number: 800 URL: https://leetcode.com/problems/find-duplicate-file-in-system Title: 609. Find Duplicate File in System Problem Description: Given a list paths of directory info, including the directory path, and all the files with contents in this directory, return all the duplicate files in the file system in terms of their paths. You may return the answer in any order. A group of duplicate files consists of at least two files that have the same content. A single directory info string in the input list has the following format:

”root/d1/d2/.../dm f1.txt(f1_content) f2.txt(f2_content) ... fn.txt(fn_content)”

It means there are n files (f1.txt, f2.txt ... fn.txt) with content (f1_content, f2_content ... fn_content) respectively in the directory ”root/d1/d2/.../dm”. Note that n >= 1 and m >= 0. If m = 0, it means the directory is just the root directory. The output is a list of groups of duplicate file paths. For each group, it contains all the file paths of the files that have the same content. A file path is a string that has the following format:

”directory_path/file_name.txt”

Example 1: Input: paths = [”root/a 1.txt(abcd) 2.txt(efgh)”,”root/c 3.txt(abcd)”,”root/c/d 4.txt(efgh)”,”root 4.txt(efgh)”] Output: [[”root/a/2.txt”,”root/c/d/4.txt”,”root/4.txt”] Example 2: Input: paths = [”root/a 1.txt(abcd) 2.txt(efgh)”,”root/c 3.txt(abcd)”,”root/c/d 4.txt(efgh)”] Output: [[”root/a/2.txt”,”root/c/d/4.txt”],[”root/a/1.txt”,”root/c/3.txt”

Constraints:

1 <= paths.length <= 2 * 104 1 <= paths[i].length <= 3000 1 <= sum(paths[i].length) <= 5 * 105 paths[i] consist of English letters, digits, ’/’, ’.’, ’(’, ’)’, and ’ ’. You may assume no files or directories share the same name in the same directory. You may assume each given directory info represents a unique directory. A single blank space separates the directory path and file info.

Follow up:

Imagine you are given a real file system, how will you search files? DFS or BFS? If the file content is very large (GB level), how will you modify your solution? If you can only read the file by 1kb each time, how will you modify your solution?

What is the time complexity of your modified solution? What is the most time-consuming part and memory-consuming part of it? How to optimize? How to make sure the duplicated files you find are not false positive?

==

Problem Number: 801 URL: https://leetcode.com/problems/valid-triangle-number Title: 611. Valid Triangle Number Problem Description: Given an integer array nums, return the number of triplets chosen from the array that can make triangles if we take them as side lengths of a triangle. Example 1: Input: nums = [2,2,3,4] Output: 3 Explanation: Valid combinations are: 2,3,4 (using the first 2) 2,3,4 (using the second 2) 2,2,3

Example 2: Input: nums = [4,2,3,4] Output: 4

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] <= 1000

==

Problem Number: 802 URL: https://leetcode.com/problems/task-scheduler Title: 621. Task Scheduler Problem Description: Given a characters array tasks, representing the tasks a CPU needs to do, where each letter represents a different task. Tasks could be done in any order. Each task is done in one unit of time. For each unit of time, the CPU could complete either one task or just be idle. However, there is a non-negative integer n that represents the cooldown period between two same tasks (the same letter in the array), that is that there must be at least n units of time between any two same tasks. Return the least number of units of times that the CPU will take to finish all the given tasks. Example 1: Input: tasks = ["A","A","A","B","B","B"], n = 2 Output: 8 Explanation: A -> B -> idle -> A -> B -> idle -> A -> B There is at least 2 units of time between any two same tasks.

Example 2: Input: tasks = ["A","A","A","B","B","B"], n = 0 Output: 6 Explanation: On this case any permutation of size 6 would work since n = 0. ["A","A","A","B","B","B"] ["A","B","A","B","A","B"] ["B","B","B","A","A","A"] ... And so on.

Example 3: Input: tasks = ["A","A","A","A","A","A","B","C","D","E","F","G"], n = 2 Output: 16 Explanation: One possible solution is A -> B -> C -> A -> D -> E -> A -> F -> G -> A -> idle -> idle -> A -> idle -> idle -> A

Constraints:

1 <= task.length <= 104 tasks[i] is upper-case English letter. The integer n is in the range [0, 100].

==

Problem Number: 803 URL: https://leetcode.com/problems/design-circular-queue Title: 622. Design Circular Queue Problem Description: Design your implementation of the circular queue. The circular queue is a linear data structure in which the operations are performed based on FIFO (First In First

Out) principle, and the last position is connected back to the first position to make a circle. It is also called ”Ring Buffer”. One of the benefits of the circular queue is that we can make use of the spaces in front of the queue. In a normal queue, once the queue becomes full, we cannot insert the next element even if there is a space in front of the queue. But using the circular queue, we can use the space to store new values. Implement the MyCircularQueue class:

MyCircularQueue(k) Initializes the object with the size of the queue to be k. int Front() Gets the front item from the queue. If the queue is empty, return -1. int Rear() Gets the last item from the queue. If the queue is empty, return -1. boolean enQueue(int value) Inserts an element into the circular queue. Return true if the operation is successful. boolean deQueue() Deletes an element from the circular queue. Return true if the operation is successful. boolean isEmpty() Checks whether the circular queue is empty or not. boolean isFull() Checks whether the circular queue is full or not.

You must solve the problem without using the built-in queue data structure in your programming language. Example 1: Input [”MyCircularQueue”, ”enQueue”, ”enQueue”, ”enQueue”, ”enQueue”, ”Rear”, ”isFull”, ”deQueue”, ”enQueue”, ”Rear”] [[3], [1], [2], [3], [4], [], [], [], [4], []] Output [null, true, true, true, false, 3, true, true, true, 4]

Explanation MyCircularQueue myCircularQueue = new MyCircularQueue(3); myCircularQueue.enQueue(1); // return True myCircularQueue.enQueue(2); // return True myCircularQueue.enQueue(3); // return True myCircularQueue.enQueue(4); // return False myCircularQueue.Rear(); // return 3 myCircularQueue.isFull(); // return True myCircularQueue.deQueue(); // return True myCircularQueue.enQueue(4); // return True myCircularQueue.Rear(); // return 4

Constraints:

1 <= k <= 1000 0 <= value <= 1000 At most 3000 calls will be made to enQueue, deQueue, Front, Rear, isEmpty, and isFull.

==

Problem Number: 804 URL: https://leetcode.com/problems/add-one-row-to-tree Title: 623. Add One Row to Tree Problem Description: Given the root of a binary tree and two integers val and depth, add a row of nodes with value val at the given depth depth. Note that the root node is at depth 1. The adding rule is:

Given the integer depth, for each not null tree node cur at the depth depth - 1, create two tree nodes with value val as cur’s left subtree root and right subtree root. cur’s original left subtree should be the left subtree of the new left subtree root. cur’s original right subtree should be the right subtree of the new right subtree root. If depth == 1 that means there is no depth depth - 1 at all, then create a tree node with value val as the new root of the whole original tree, and the original tree is the new root’s left subtree.

Example 1:

Input: root = [4,2,6,3,1,5], val = 1, depth = 2 Output: [4,1,1,2,null,null,6,3,1,5]

Example 2:

Input: root = [4,2,null,3,1], val = 1, depth = 3 Output: [4,2,null,1,1,3,null,null,1]

Constraints:

The number of nodes in the tree is in the range [1, 104]. The depth of the tree is in the range [1, 104]. -100 <= Node.val <= 100 -105 <= val <= 105 1 <= depth <= the depth of tree + 1

==
Problem Number: 805 URL: https://leetcode.com/problems/sum-of-square-numbers Title: 633. Sum of Square Numbers Problem Description: Given a non-negative integer c, decide whether there're two integers a and b such that a2 + b2 = c. Example 1: Input: c = 5 Output: true Explanation: 1 * 1 + 2 * 2 = 5

Example 2: Input: c = 3 Output: false

Constraints:

0 <= c <= 231 - 1

==
Problem Number: 806 URL: https://leetcode.com/problems/exclusive-time-of-functions Title: 636. Exclusive Time of Functions Problem Description: On a single-threaded CPU, we execute a program containing n functions. Each function has a unique ID between 0 and n-1. Function calls are stored in a call stack: when a function call starts, its ID is pushed onto the stack, and when a function call ends, its ID is popped off the stack. The function whose ID is at the top of the stack is the current function being executed. Each time a function starts or ends, we write a log with the ID, whether it started or ended, and the timestamp. You are given a list logs, where logs[i] represents the ith log message formatted as a string "{function_id}:{"start" | "end"}:{timestamp}". For example, "0:start:3" means a function call with function ID 0 started at the beginning of timestamp 3, and "1:end:2" means a function call with function ID 1 ended at the end of timestamp 2. Note that a function can be called multiple times, possibly recursively. A function's exclusive time is the sum of execution times for all function calls in the program. For example, if a function is called twice, one call executing for 2 time units and another call executing for 1 time unit, the exclusive time is 2 + 1 = 3. Return the exclusive time of each function in an array, where the value at the ith index represents the exclusive time for the function with ID i. Example 1:

Input: n = 2, logs = ["0:start:0","1:start:2","1:end:5","0:end:6"] Output: [3,4] Explanation: Function 0 starts at the beginning of time 0, then it executes 2 for units of time and reaches the end of time 1. Function 1 starts at the beginning of time 2, executes for 4 units of time, and ends at the end of time 5. Function 0 resumes execution at the beginning of time 6 and executes for 1 unit of time. So function 0 spends 2 + 1 = 3 units of total time executing, and function 1 spends 4 units of total time executing.

Example 2: Input: n = 1, logs = ["0:start:0","0:start:2","0:end:5","0:start:6","0:end:6","0:end:7"] Output: [8] Explanation: Function 0 starts at the beginning of time 0, executes for 2 units of time, and recursively calls itself. Function 0 (recursive call) starts at the beginning of time 2 and executes for 4 units of time. Function 0 (initial call) resumes execution then immediately calls itself again. Function 0 (2nd recursive call) starts at the beginning of time 6 and executes for 1 unit of time. Function 0 (initial call) resumes execution at the beginning of time 7 and executes for 1 unit of time. So function 0 spends 2 + 4 + 1 + 1 = 8 units of total time executing.

Example 3: Input: n = 2, logs = ["0:start:0","0:start:2","0:end:5","1:start:6","1:end:6","0:end:7"] Output: [7,1] Explanation: Function 0 starts at the beginning of time 0, executes for 2 units of time, and recursively calls itself. Function 0 (recursive call) starts at the beginning of time 2 and executes for 4 units of time. Function 0 (initial call) resumes execution then immediately calls function 1. Function 1 starts at the beginning of time 6, executes 1 unit of time, and ends at the end of time 6. Function 0 resumes execution at the beginning of time 6 and executes for 2 units of time. So function 0 spends 2 + 4 + 1 = 7 units of total time executing, and function 1 spends 1 unit of total time executing.

Constraints:

1 <= n <= 100 1 <= logs.length <= 500 0 <= function_id < n 0 <= timestamp <= 109 No two start events will happen at the same timestamp. No two end events will happen at the same timestamp. Each function has an "end" log for each "start" log.

==

Problem Number: 807 URL: https://leetcode.com/problems/shopping-offers Title: 638. Shopping Offers Problem Description: In LeetCode Store, there are n items to sell. Each item has a price. However, there are some special offers, and a special offer consists of one or more different kinds of items with a sale price. You are given an integer array price where price[i] is the price of the ith item, and an integer array needs where needs[i] is the number of pieces of the ith item you want to buy. You are also given an array special where special[i] is of size n + 1 where special[i][j] is the number of pieces of the jth item in the ith offer and special[i][n] (i.e., the last integer in the array) is the price of the ith offer. Return the lowest price you have to pay for exactly certain items as given, where you could make optimal use of the special offers. You are not allowed to buy more items than you want, even if that would lower the overall price. You could use any of the special offers as many times as you want. Example 1: Input: price = [2,5], special = [[3,0,5],[1,2,10]], needs = [3,2] Output: 14 Explanation: There are two kinds of items, A and B. Their prices are $2 and $5 respectively. In special offer 1, you can pay $5 for 3A and 0B In special offer 2, you can pay $10 for 1A and 2B. You need to buy 3A and 2B, so you may pay $10 for 1A and 2B (special offer #2), and $4 for 2A.

Example 2: Input: price = [2,3,4], special = [[1,1,0,4],[2,2,1,9]], needs = [1,2,1] Output: 11 Explanation: The price of A is $2, and $3 for B, $4 for C. You may pay $4 for 1A and 1B, and $9 for 2A ,2B and 1C. You need to buy 1A ,2B and 1C, so you may pay $4 for 1A and 1B (special offer #1), and $3 for 1B, $4 for 1C. You cannot add more items, though only $9 for 2A ,2B and 1C.

Constraints:

n == price.length == needs.length 1 <= n <= 6 0 <= price[i], needs[i] <= 10 1 <= special.length <= 100 special[i].length == n + 1 0 <= special[i][j] <= 50

==
Problem Number: 808 URL: https://leetcode.com/problems/solve-the-equation Title: 640. Solve the Equation Problem Description: Solve a given equation and return the value of 'x' in the form of a string "x=#value". The equation contains only '+', '-' operation, the variable 'x' and its coefficient. You should return "No solution" if there is no solution for the equation, or "Infinite solutions" if there are infinite solutions for the equation. If there is exactly one solution for the equation, we ensure that the value of 'x' is an integer. Example 1: Input: equation = "x+5-3+x=6+x-2" Output: "x=2"

Example 2: Input: equation = "x=x" Output: "Infinite solutions"

Example 3: Input: equation = "2x=x" Output: "x=0"

Constraints:

3 <= equation.length <= 1000 equation has exactly one '='. equation consists of integers with an absolute value in the range [0, 100] without any leading zeros, and the variable 'x'.

==
Problem Number: 809 URL: https://leetcode.com/problems/design-circular-deque Title: 641. Design Circular Deque Problem Description: Design your implementation of the circular double-ended queue (deque). Implement the MyCircularDeque class:

MyCircularDeque(int k) Initializes the deque with a maximum size of k. boolean insertFront() Adds an item at the front of Deque. Returns true if the operation is successful, or false otherwise. boolean insertLast() Adds an item at the rear of Deque. Returns true if the operation is successful, or false otherwise. boolean deleteFront() Deletes an item from the front of Deque. Returns true if the operation is successful, or false otherwise. boolean deleteLast() Deletes an item from the rear of Deque. Returns true if the operation is successful, or false otherwise. int getFront() Returns the front item from the Deque. Returns -1 if the deque is empty. int getRear() Returns the last item from Deque. Returns -1 if the deque is empty. boolean isEmpty() Returns true if the deque is empty, or false otherwise. boolean isFull() Returns true if the deque is full, or false otherwise.

Example 1: Input ["MyCircularDeque", "insertLast", "insertLast", "insertFront", "insertFront", "getRear", "isFull", "deleteLast", "insertFront", "getFront"] [[3], [1], [2], [3], [4], [], [], [], [4], []] Output [null, true, true, true, false, 2, true, true, true, 4]

Explanation MyCircularDeque myCircularDeque = new MyCircularDeque(3); myCircularDeque.insertLast(1); // return True myCircularDeque.insertLast(2); // return True myCircularDeque.insertFront(3); // return True myCircularDeque.insertFront(4); // return False, the queue is full. myCircularDeque.getRear(); // return 2 myCircularDeque.isFull(); // return True myCircularDeque.deleteLast(); // return True myCircularDeque.insertFront(4); // return True myCircularDeque.getFront(); // return 4

Constraints:

1 <= k <= 1000 0 <= value <= 1000 At most 2000 calls will be made to insertFront, insertLast, deleteFront, deleteLast, getFront, getRear, isEmpty, isFull.

==
Problem Number: 810 URL: https://leetcode.com/problems/maximum-length-of-pair-chain Title: 646. Maximum Length of Pair Chain Problem Description: You are given an array of n pairs pairs where pairs[i] = [lefti, righti] and lefti < righti. A pair p2 = [c, d] follows a pair p1 = [a, b] if b < c. A chain of pairs can be formed in this fashion. Return the length longest chain which can be formed. You do not need to use up all the given intervals. You can select pairs in any order. Example 1: Input: pairs = [[1,2],[2,3],[3,4]] Output: 2 Explanation: The longest chain is [1,2] -> [3,4].

Example 2: Input: pairs = [[1,2],[7,8],[4,5]] Output: 3 Explanation: The longest chain is [1,2] -> [4,5] -> [7,8].

Constraints:

n == pairs.length 1 <= n <= 1000 -1000 <= lefti < righti <= 1000

==
Problem Number: 811 URL: https://leetcode.com/problems/palindromic-substrings Title: 647. Palindromic Substrings Problem Description: Given a string s, return the number of palindromic substrings in it. A string is a palindrome when it reads the same backward as forward. A substring is a contiguous sequence of characters within the string. Example 1: Input: s = "abc" Output: 3 Explanation: Three palindromic strings: "a", "b", "c".

Example 2: Input: s = "aaa" Output: 6 Explanation: Six palindromic strings: "a", "a", "a", "aa", "aa", "aaa".

Constraints:

1 <= s.length <= 1000 s consists of lowercase English letters.

===

Problem Number: 812 URL: https://leetcode.com/problems/replace-words Title: 648. Replace Words Problem Description: In English, we have a concept called root, which can be followed by some other word to form another longer word - let's call this word successor. For example, when the root "an" is followed by the successor word "other", we can form a new word "another". Given a dictionary consisting of many roots and a sentence consisting of words separated by spaces, replace all the successors in the sentence with the root forming it. If a successor can be replaced by more than one root, replace it with the root that has the shortest length. Return the sentence after the replacement. Example 1: Input: dictionary = ["cat","bat","rat"], sentence = "the cattle was rattled by the battery" Output: "the cat was rat by the bat"

Example 2: Input: dictionary = ["a","b","c"], sentence = "aadsfasf absbs bbab cadsfafs" Output: "a a b c"

Constraints:

1 <= dictionary.length <= 1000 1 <= dictionary[i].length <= 100 dictionary[i] consists of only lower-case letters. 1 <= sentence.length <= 106 sentence consists of only lower-case letters and spaces. The number of words in sentence is in the range [1, 1000] The length of each word in sentence is in the range [1, 1000] Every two consecutive words in sentence will be separated by exactly one space. sentence does not have leading or trailing spaces.

===

Problem Number: 813 URL: https://leetcode.com/problems/dota2-senate Title: 649. Dota2 Senate Problem Description: In the world of Dota2, there are two parties: the Radiant and the Dire. The Dota2 senate consists of senators coming from two parties. Now the Senate wants to decide on a change in the Dota2 game. The voting for this change is a round-based procedure. In each round, each senator can exercise one of the two rights:

Ban one senator's right: A senator can make another senator lose all his rights in this and all the following rounds. Announce the victory: If this senator found the senators who still have rights to vote are all from the same party, he can announce the victory and decide on the change in the game.

Given a string senate representing each senator's party belonging. The character 'R' and 'D' represent the Radiant party and the Dire party. Then if there are n senators, the size of the given string will be n. The round-based procedure starts from the first senator to the last senator in the given order. This procedure will last until the end of voting. All the senators who have lost their rights will be skipped during the procedure. Suppose every senator is smart enough and will play the best strategy for his own party. Predict which party will finally announce the victory and change the Dota2 game. The output should be "Radiant" or "Dire". Example 1: Input: senate = "RD" Output: "Radiant" Explanation: The first senator comes from Radiant and he can just ban the next senator's right in round 1. And the second senator can't exercise any rights anymore since his right has been banned. And in round 2, the first senator can just announce the victory since he is the only guy in the senate who can vote.

Example 2: Input: senate = ”RDD” Output: ”Dire” Explanation: The first senator comes from Radiant and he can just ban the next senator’s right in round 1. And the second senator can’t exercise any rights anymore since his right has been banned. And the third senator comes from Dire and he can ban the first senator’s right in round 1. And in round 2, the third senator can just announce the victory since he is the only guy in the senate who can vote.

Constraints:

n == senate.length 1 <= n <= 104 senate[i] is either ’R’ or ’D’.

==

Problem Number: 814 URL: https://leetcode.com/problems/2-keys-keyboard Title: 650. 2 Keys Keyboard Problem Description: There is only one character ’A’ on the screen of a notepad. You can perform one of two operations on this notepad for each step:

Copy All: You can copy all the characters present on the screen (a partial copy is not allowed). Paste: You can paste the characters which are copied last time.

Given an integer n, return the minimum number of operations to get the character ’A’ exactly n times on the screen. Example 1: Input: n = 3 Output: 3 Explanation: Initially, we have one character ’A’. In step 1, we use Copy All operation. In step 2, we use Paste operation to get ’AA’. In step 3, we use Paste operation to get ’AAA’.

Example 2: Input: n = 1 Output: 0

Constraints:

1 <= n <= 1000

==

Problem Number: 815 URL: https://leetcode.com/problems/find-duplicate-subtrees Title: 652. Find Duplicate Subtrees Problem Description: Given the root of a binary tree, return all duplicate subtrees. For each kind of duplicate subtrees, you only need to return the root node of any one of them. Two trees are duplicate if they have the same structure with the same node values. Example 1:

Input: root = [1,2,3,4,null,2,4,null,null,4] Output: [[2,4],[4]]

Example 2:

Input: root = [2,1,1] Output: [[1]]

Example 3:

Input: root = [2,2,2,3,null,3,null] Output: [[2,3],[3]]

Constraints:

The number of the nodes in the tree will be in the range [1, 5000] -200 <= Node.val <= 200

==
Problem Number: 816 URL: https://leetcode.com/problems/maximum-binary-tree Title: 654. Maximum Binary Tree Problem Description: You are given an integer array nums with no duplicates. A maximum binary tree can be built recursively from nums using the following algorithm:

Create a root node whose value is the maximum value in nums. Recursively build the left subtree on the subarray prefix to the left of the maximum value. Recursively build the right subtree on the subarray suffix to the right of the maximum value.

Return the maximum binary tree built from nums.   Example 1:

Input: nums = [3,2,1,6,0,5] Output: [6,3,5,null,2,0,null,null,1] Explanation: The recursive calls are as follow: - The largest value in [3,2,1,6,0,5] is 6. Left prefix is [3,2,1] and right suffix is [0,5]. - The largest value in [3,2,1] is 3. Left prefix is [] and right suffix is [2,1]. - Empty array, so no child. - The largest value in [2,1] is 2. Left prefix is [] and right suffix is [1]. - Empty array, so no child. - Only one element, so child is a node with value 1. - The largest value in [0,5] is 5. Left prefix is [0] and right suffix is []. - Only one element, so child is a node with value 0. - Empty array, so no child.

Example 2:

Input: nums = [3,2,1] Output: [3,null,2,null,1]

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] <= 1000 All integers in nums are unique.

==
Problem Number: 817 URL: https://leetcode.com/problems/print-binary-tree Title: 655. Print Binary Tree Problem Description: Given the root of a binary tree, construct a 0-indexed m x n string matrix res that represents a formatted layout of the tree. The formatted layout matrix should be constructed using the following rules:

The height of the tree is height and the number of rows m should be equal to height + 1. The number of columns n should be equal to 2height+1 - 1. Place the root node in the middle of the top row (more formally, at location res[0][(n-1)/2]). For each node that has been placed in the matrix at position res[r][c], place its left child at res[r+1][c-2height-r-1] and its right child at res[r+1][c+2height-r-1]. Continue this process until all the nodes in the tree have been placed. Any empty cells should contain the empty string "".

Return the constructed matrix res.   Example 1:

Input: root = [1,2] Output: [["","1",""], ["2","",""]]

Example 2:

Input: root = [1,2,3,null,4] Output: [["","","","1","","",""], ["","2","","","","3",""], ["","","4","","","",""]]

Constraints:

The number of nodes in the tree is in the range [1, 210]. -99 <= Node.val <= 99 The depth of the tree will be in the range [1, 10].

==
Problem Number: 818 URL: https://leetcode.com/problems/find-k-closest-elements Title: 658. Find K Closest Elements Problem Description: Given a sorted integer array arr, two integers k and x, return the k closest integers to x in the array. The result should also be sorted in ascending order. An integer a is closer to x than an integer b if:

|a - x| < |b - x|, or |a - x| == |b - x| and a < b

Example 1: Input: arr = [1,2,3,4,5], k = 4, x = 3 Output: [1,2,3,4] Example 2: Input: arr = [1,2,3,4,5], k = 4, x = -1 Output: [1,2,3,4]

Constraints:

1 <= k <= arr.length 1 <= arr.length <= 104 arr is sorted in ascending order. -104 <= arr[i], x <= 104

==
Problem Number: 819 URL: https://leetcode.com/problems/split-array-into-consecutive-subsequences Title: 659. Split Array into Consecutive Subsequences Problem Description: You are given an integer array nums that is sorted in non-decreasing order. Determine if it is possible to split nums into one or more subsequences such that both of the following conditions are true:

Each subsequence is a consecutive increasing sequence (i.e. each integer is exactly one more than the previous integer). All subsequences have a length of 3 or more.

Return true if you can split nums according to the above conditions, or false otherwise. A subsequence of an array is a new array that is formed from the original array by deleting some (can be none) of the elements without disturbing the relative positions of the remaining elements. (i.e., [1,3,5] is a subsequence of [1,2,3,4,5] while [1,3,2] is not). Example 1: Input: nums = [1,2,3,3,4,5] Output: true Explanation: nums can be split into the following subsequences: [1,2,3,3,4,5] --> 1, 2, 3 [1,2,3,3,4,5] --> 3, 4, 5

Example 2: Input: nums = [1,2,3,3,4,4,5,5] Output: true Explanation: nums can be split into the following subsequences: [1,2,3,3,4,4,5,5] --> 1, 2, 3, 4, 5 [1,2,3,3,4,4,5,5] --> 3, 4, 5

Example 3: Input: nums = [1,2,3,4,4,5] Output: false Explanation: It is impossible to split nums into consecutive increasing subsequences of length 3 or more.

Constraints:

1 <= nums.length <= 104 -1000 <= nums[i] <= 1000 nums is sorted in non-decreasing order.

==

Problem Number: 820 URL: https://leetcode.com/problems/maximum-width-of-binary-tree Title: 662. Maximum Width of Binary Tree Problem Description: Given the root of a binary tree, return the maximum width of the given tree. The maximum width of a tree is the maximum width among all levels. The width of one level is defined as the length between the end-nodes (the leftmost and rightmost non-null nodes), where the null nodes between the end-nodes that would be present in a complete binary tree extending down to that level are also counted into the length calculation. It is guaranteed that the answer will in the range of a 32-bit signed integer. Example 1:

Input: root = [1,3,2,5,3,null,9] Output: 4 Explanation: The maximum width exists in the third level with length 4 (5,3,null,9).

Example 2:

Input: root = [1,3,2,5,null,null,9,6,null,7] Output: 7 Explanation: The maximum width exists in the fourth level with length 7 (6,null,null,null,null,null,7).

Example 3:

Input: root = [1,3,2,5] Output: 2 Explanation: The maximum width exists in the second level with length 2 (3,2).

Constraints:

The number of nodes in the tree is in the range [1, 3000]. -100 <= Node.val <= 100

==

Problem Number: 821 URL: https://leetcode.com/problems/non-decreasing-array Title: 665. Non-decreasing Array Problem Description: Given an array nums with n integers, your task is to check if it could become non-decreasing by modifying at most one element. We define an array is non-decreasing if nums[i] <= nums[i + 1] holds for every i (0-based) such that (0 <= i <= n - 2). Example 1: Input: nums = [4,2,3] Output: true Explanation: You could modify the first 4 to 1 to get a non-decreasing array.

Example 2: Input: nums = [4,2,1] Output: false Explanation: You cannot get a non-decreasing array by modifying at most one element.

Constraints:

n == nums.length 1 <= n <= 104 -105 <= nums[i] <= 105

==
Problem Number: 822 URL: https://leetcode.com/problems/beautiful-arrangement-ii Title: 667. Beautiful Arrangement II Problem Description: Given two integers n and k, construct a list answer that contains n different positive integers ranging from 1 to n and obeys the following requirement:

Suppose this list is answer = [a1, a2, a3, ... , an], then the list [|a1 - a2|, |a2 - a3|, |a3 - a4|, ... , |an-1 - an|] has exactly k distinct integers.

Return the list answer. If there multiple valid answers, return any of them. Example 1: Input: n = 3, k = 1 Output: [1,2,3] Explanation: The [1,2,3] has three different positive integers ranging from 1 to 3, and the [1,1] has exactly 1 distinct integer: 1

Example 2: Input: n = 3, k = 2 Output: [1,3,2] Explanation: The [1,3,2] has three different positive integers ranging from 1 to 3, and the [2,1] has exactly 2 distinct integers: 1 and 2.

Constraints:

1 <= k < n <= 104

==
Problem Number: 823 URL: https://leetcode.com/problems/trim-a-binary-search-tree Title: 669. Trim a Binary Search Tree Problem Description: Given the root of a binary search tree and the lowest and highest boundaries as low and high, trim the tree so that all its elements lies in [low, high]. Trimming the tree should not change the relative structure of the elements that will remain in the tree (i.e., any node's descendant should remain a descendant). It can be proven that there is a unique answer. Return the root of the trimmed binary search tree. Note that the root may change depending on the given bounds. Example 1:

Input: root = [1,0,2], low = 1, high = 2 Output: [1,null,2]

Example 2:

Input: root = [3,0,4,null,2,null,null,1], low = 1, high = 3 Output: [3,2,null,1]

Constraints:

The number of nodes in the tree is in the range [1, 104]. 0 <= Node.val <= 104 The value of each node in the tree is unique. root is guaranteed to be a valid binary search tree. 0 <= low <= high <= 104

==
Problem Number: 824 URL: https://leetcode.com/problems/maximum-swap Title: 670. Maximum Swap Problem Description: You are given an integer num. You can swap two digits at most once to get the maximum valued number. Return the maximum valued number you can get. Example 1: Input: num = 2736 Output: 7236 Explanation: Swap the number 2 and the number 7.

Example 2: Input: num = 9973 Output: 9973 Explanation: No swap.

Constraints:

0 <= num <= 108

==

Problem Number: 825 URL: https://leetcode.com/problems/bulb-switcher-ii Title: 672. Bulb Switcher II Problem Description: There is a room with n bulbs labeled from 1 to n that all are turned on initially, and four buttons on the wall. Each of the four buttons has a different functionality where:

Button 1: Flips the status of all the bulbs. Button 2: Flips the status of all the bulbs with even labels (i.e., 2, 4, ...). Button 3: Flips the status of all the bulbs with odd labels (i.e., 1, 3, ...). Button 4: Flips the status of all the bulbs with a label j = 3k + 1 where k = 0, 1, 2, ... (i.e., 1, 4, 7, 10, ...).

You must make exactly presses button presses in total. For each press, you may pick any of the four buttons to press. Given the two integers n and presses, return the number of different possible statuses after performing all presses button presses. Example 1: Input: n = 1, presses = 1 Output: 2 Explanation: Status can be: - [off] by pressing button 1 - [on] by pressing button 2

Example 2: Input: n = 2, presses = 1 Output: 3 Explanation: Status can be: - [off, off] by pressing button 1 - [on, off] by pressing button 2 - [off, on] by pressing button 3

Example 3: Input: n = 3, presses = 1 Output: 4 Explanation: Status can be: - [off, off, off] by pressing button 1 - [off, on, off] by pressing button 2 - [on, off, on] by pressing button 3 - [off, on, on] by pressing button 4

Constraints:

1 <= n <= 1000 0 <= presses <= 1000

==

Problem Number: 826 URL: https://leetcode.com/problems/number-of-longest-increasing-subsequence Title: 673. Number of Longest Increasing Subsequence Problem Description: Given an integer array nums, return the number of longest increasing subsequences. Notice that the sequence has to be strictly increasing. Example 1: Input: nums = [1,3,5,4,7] Output: 2 Explanation: The two longest increasing subsequences are [1, 3, 4, 7] and [1, 3, 5, 7].

Example 2: Input: nums = [2,2,2,2,2] Output: 5 Explanation: The length of the longest increasing subsequence is 1, and there are 5 increasing subsequences of length 1, so output 5.

Constraints:

1 <= nums.length <= 2000 -106 <= nums[i] <= 106

==
Problem Number: 827 URL: https://leetcode.com/problems/implement-magic-dictionary Title: 676. Implement Magic Dictionary Problem Description: Design a data structure that is initialized with a list of different words. Provided a string, you should determine if you can change exactly one character in this string to match any word in the data structure. Implement the MagicDictionary class:

MagicDictionary() Initializes the object. void buildDict(String[] dictionary) Sets the data structure with an array of distinct strings dictionary. bool search(String searchWord) Returns true if you can change exactly one character in searchWord to match any string in the data structure, otherwise returns false.

Example 1: Input ["MagicDictionary", "buildDict", "search", "search", "search", "search"] [[], [["hello", "leetcode"]], ["hello"], ["hhllo"], ["hell"], ["leetcoded"]] Output [null, null, false, true, false, false]

Explanation MagicDictionary magicDictionary = new MagicDictionary(); magicDictionary.buildDict(["hello", "leetcode"]); magicDictionary.search("hello"); // return False magicDictionary.search("hhllo"); // We can change the second 'h' to 'e' to match "hello" so we return True magicDictionary.search("hell"); // return False magicDictionary.search("leetcoded"); // return False

Constraints:

1 <= dictionary.length <= 100 1 <= dictionary[i].length <= 100 dictionary[i] consists of only lower-case English letters. All the strings in dictionary are distinct. 1 <= searchWord.length <= 100 searchWord consists of only lower-case English letters. buildDict will be called only once before search. At most 100 calls will be made to search.

==
Problem Number: 828 URL: https://leetcode.com/problems/map-sum-pairs Title: 677. Map Sum Pairs Problem Description: Design a map that allows you to do the following:

Maps a string key to a given value. Returns the sum of the values that have a key with a prefix equal to a given string.

Implement the MapSum class:

MapSum() Initializes the MapSum object. void insert(String key, int val) Inserts the key-val pair into the map. If the key already existed, the original key-value pair will be overridden to the new one. int sum(string prefix) Returns the sum of all the pairs' value whose key starts with the prefix.

Example 1: Input ["MapSum", "insert", "sum", "insert", "sum"] [[], ["apple", 3], ["ap"], ["app", 2], ["ap"]] Output [null, null, 3, null, 5]

Explanation MapSum mapSum = new MapSum(); mapSum.insert("apple", 3);

mapSum.sum(”ap”); // return 3 (apple = 3) mapSum.insert(”app”, 2); mapSum.sum(”ap”); // return 5 (apple + app = 3 + 2 = 5)

Constraints:

1 <= key.length, prefix.length <= 50 key and prefix consist of only lowercase English letters. 1 <= val <= 1000 At most 50 calls will be made to insert and sum.

==
Problem Number: 829 URL: https://leetcode.com/problems/valid-parenthesis-string Title: 678. Valid Parenthesis String Problem Description: Given a string s containing only three types of characters: ’(’, ’)’ and ’*’, return true if s is valid. The following rules define a valid string:

Any left parenthesis ’(’ must have a corresponding right parenthesis ’)’. Any right parenthesis ’)’ must have a corresponding left parenthesis ’(’. Left parenthesis ’(’ must go before the corresponding right parenthesis ’)’. ’*’ could be treated as a single right parenthesis ’)’ or a single left parenthesis ’(’ or an empty string ””.

Example 1: Input: s = ”()” Output: true Example 2: Input: s = ”(*)” Output: true Example 3: Input: s = ”(*))” Output: true

Constraints:

1 <= s.length <= 100 s[i] is ’(’, ’)’ or ’*’.

==
Problem Number: 830 URL: https://leetcode.com/problems/redundant-connection Title: 684. Redundant Connection Problem Description: In this problem, a tree is an undirected graph that is connected and has no cycles. You are given a graph that started as a tree with n nodes labeled from 1 to n, with one additional edge added. The added edge has two different vertices chosen from 1 to n, and was not an edge that already existed. The graph is represented as an array edges of length n where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the graph. Return an edge that can be removed so that the resulting graph is a tree of n nodes. If there are multiple answers, return the answer that occurs last in the input. Example 1:

Input: edges = [[1,2],[1,3],[2,3]] Output: [2,3]

Example 2:

Input: edges = [[1,2],[2,3],[3,4],[1,4],[1,5]] Output: [1,4]

Constraints:

n == edges.length 3 <= n <= 1000 edges[i].length == 2 1 <= ai < bi <= edges.length ai != bi There are no repeated edges. The given graph is connected.

==
Problem Number: 831 URL: https://leetcode.com/problems/repeated-string-

match Title: 686. Repeated String Match Problem Description: Given two strings a and b, return the minimum number of times you should repeat string a so that string b is a substring of it. If it is impossible for b to be a substring of a after repeating it, return -1. Notice: string "abc" repeated 0 times is "", repeated 1 time is "abc" and repeated 2 times is "abcabc". Example 1: Input: a = "abcd", b = "cdabcdab" Output: 3 Explanation: We return 3 because by repeating a three times "abcdabcdabcd", b is a substring of it.

Example 2: Input: a = "a", b = "aa" Output: 2

Constraints:

1 <= a.length, b.length <= 104 a and b consist of lowercase English letters.

==
Problem Number: 832 URL: https://leetcode.com/problems/longest-univalue-path Title: 687. Longest Univalue Path Problem Description: Given the root of a binary tree, return the length of the longest path, where each node in the path has the same value. This path may or may not pass through the root. The length of the path between two nodes is represented by the number of edges between them. Example 1:

Input: root = [5,4,5,1,1,null,5] Output: 2 Explanation: The shown image shows that the longest path of the same value (i.e. 5).

Example 2:

Input: root = [1,4,5,4,4,null,5] Output: 2 Explanation: The shown image shows that the longest path of the same value (i.e. 4).

Constraints:

The number of nodes in the tree is in the range [0, 104]. -1000 <= Node.val <= 1000 The depth of the tree will not exceed 1000.

==
Problem Number: 833 URL: https://leetcode.com/problems/knight-probability-in-chessboard Title: 688. Knight Probability in Chessboard Problem Description: On an n x n chessboard, a knight starts at the cell (row, column) and attempts to make exactly k moves. The rows and columns are 0-indexed, so the top-left cell is (0, 0), and the bottom-right cell is (n - 1, n - 1). A chess knight has eight possible moves it can make, as illustrated below. Each move is two cells in a cardinal direction, then one cell in an orthogonal direction.

Each time the knight is to move, it chooses one of eight possible moves uniformly at random (even if the piece would go off the chessboard) and moves there. The knight continues moving until it has made exactly k moves or has moved off the chessboard. Return the probability that the knight remains on the board after it has stopped moving. Example 1: Input: n = 3, k = 2, row = 0, column = 0 Output: 0.06250 Explanation: There are two moves (to (1,2), (2,1)) that will keep the knight on the board. From each of those positions, there are also two moves that will keep the knight on the board. The total probability the knight stays on the board is 0.0625.

Example 2: Input: n = 1, k = 0, row = 0, column = 0 Output: 1.00000

Constraints:

1 <= n <= 25 0 <= k <= 100 0 <= row, column <= n - 1

==
Problem Number: 834 URL: https://leetcode.com/problems/employee-importance Title: 690. Employee Importance Problem Description: You have a data structure of employee information, including the employee's unique ID, importance value, and direct subordinates' IDs. You are given an array of employees employees where:

employees[i].id is the ID of the ith employee. employees[i].importance is the importance value of the ith employee. employees[i].subordinates is a list of the IDs of the direct subordinates of the ith employee.

Given an integer id that represents an employee's ID, return the total importance value of this employee and all their direct and indirect subordinates. Example 1:

Input: employees = [[1,5,[2,3]],[2,3,[]],[3,3,[]]], id = 1 Output: 11 Explanation: Employee 1 has an importance value of 5 and has two direct subordinates: employee 2 and employee 3. They both have an importance value of 3. Thus, the total importance value of employee 1 is 5 + 3 + 3 = 11.

Example 2:

Input: employees = [[1,2,[5]],[5,-3,[]]], id = 5 Output: -3 Explanation: Employee 5 has an importance value of -3 and has no direct subordinates. Thus, the total importance value of employee 5 is -3.

Constraints:

1 <= employees.length <= 2000 1 <= employees[i].id <= 2000 All employees[i].id are unique. -100 <= employees[i].importance <= 100 One employee has at most one direct leader and may have several subordinates. The IDs in employees[i].subordinates are valid IDs.

==
Problem Number: 835 URL: https://leetcode.com/problems/top-k-frequent-words Title: 692. Top K Frequent Words Problem Description: Given an array of strings words and an integer k, return the k most frequent strings. Return the answer sorted by the frequency from highest to lowest. Sort the words with the same frequency by their lexicographical order. Example 1: Input: words = ["i","love","leetcode","i","love","coding"], k = 2 Output: ["i","love"] Explanation: "i" and "love" are the two most frequent words. Note that "i" comes before "love" due to a lower alphabetical order.

Example 2: Input: words = ["the","day","is","sunny","the","the","the","sunny","is","is"], k = 4 Output: ["the","is","sunny","day"] Explanation: "the", "is", "sunny" and "day" are the four most frequent words, with the number of occurrence being 4, 3, 2 and 1 respectively.

Constraints:

1 <= words.length <= 500 1 <= words[i].length <= 10 words[i] consists of lowercase English letters. k is in the range [1, The number of unique words[i]]

Follow-up: Could you solve it in O(n log(k)) time and O(n) extra space?

==
Problem Number: 836 URL: https://leetcode.com/problems/max-area-of-island Title: 695. Max Area of Island Problem Description: You are given an m x n binary matrix grid. An island is a group of 1's (representing land) connected 4-directionally (horizontal or vertical.) You may assume all four edges of the grid are surrounded by water. The area of an island is the number of cells with a value 1 in the island. Return the maximum area of an island in grid. If there is no island, return 0. Example 1:

Input: grid = [[0,0,1,0,0,0,0,1,0,0,0,0,0],[0,0,0,0,0,0,0,1,1,1,0,0,0],[0,1,1,0,1,0,0,0,0,0,0,0,0],[0,1,0,0,1,1,0,0,1,0,1,0,
Output: 6 Explanation: The answer is not 11, because the island must be connected 4-directionally.

Example 2: Input: grid = [[0,0,0,0,0,0,0,0]] Output: 0

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 50 grid[i][j] is either 0 or 1.

==
Problem Number: 837 URL: https://leetcode.com/problems/partition-to-k-equal-sum-subsets Title: 698. Partition to K Equal Sum Subsets Problem Description: Given an integer array nums and an integer k, return true if it is possible to divide this array into k non-empty subsets whose sums are all equal. Example 1: Input: nums = [4,3,2,3,5,2,1], k = 4 Output: true Explanation: It is possible to divide it into 4 subsets (5), (1, 4), (2,3), (2,3) with equal sums.

Example 2: Input: nums = [1,2,3,4], k = 3 Output: false

Constraints:

1 <= k <= nums.length <= 16 1 <= nums[i] <= 104 The frequency of each element is in the range [1, 4].

==
Problem Number: 838 URL: https://leetcode.com/problems/insert-into-a-binary-search-tree Title: 701. Insert into a Binary Search Tree Problem Description: You are given the root node of a binary search tree (BST) and a value to insert into the tree. Return the root node of the BST after the

insertion. It is guaranteed that the new value does not exist in the original BST. Notice that there may exist multiple valid ways for the insertion, as long as the tree remains a BST after insertion. You can return any of them. Example 1:

Input: root = [4,2,7,1,3], val = 5 Output: [4,2,7,1,3,5] Explanation: Another accepted tree is:

Example 2: Input: root = [40,20,60,10,30,50,70], val = 25 Output: [40,20,60,10,30,50,70,null,null,25]

Example 3: Input: root = [4,2,7,1,3,null,null,null,null,null,null], val = 5 Output: [4,2,7,1,3,5]

Constraints:

The number of nodes in the tree will be in the range [0, 104]. -108 <= Node.val <= 108 All the values Node.val are unique. -108 <= val <= 108 It's guaranteed that val does not exist in the original BST.

==
Problem Number: 839 URL: https://leetcode.com/problems/design-linked-list Title: 707. Design Linked List Problem Description: Design your implementation of the linked list. You can choose to use a singly or doubly linked list. A node in a singly linked list should have two attributes: val and next. val is the value of the current node, and next is a pointer/reference to the next node. If you want to use the doubly linked list, you will need one more attribute prev to indicate the previous node in the linked list. Assume all nodes in the linked list are 0-indexed. Implement the MyLinkedList class:

MyLinkedList() Initializes the MyLinkedList object. int get(int index) Get the value of the indexth node in the linked list. If the index is invalid, return -1. void addAtHead(int val) Add a node of value val before the first element of the linked list. After the insertion, the new node will be the first node of the linked list. void addAtTail(int val) Append a node of value val as the last element of the linked list. void addAtIndex(int index, int val) Add a node of value val before the indexth node in the linked list. If index equals the length of the linked list, the node will be appended to the end of the linked list. If index is greater than the length, the node will not be inserted. void deleteAtIndex(int index) Delete the indexth node in the linked list, if the index is valid.

Example 1: Input ["MyLinkedList", "addAtHead", "addAtTail", "addAtIndex", "get", "deleteAtIndex", "get"] [[], [1], [3], [1, 2], [1], [1], [1]] Output [null, null, null, null, 2, null, 3]

Explanation MyLinkedList myLinkedList = new MyLinkedList(); myLinkedList.addAtHead(1); myLinkedList.addAtTail(3); myLinkedList.addAtIndex(1, 2); // linked list becomes 1->2->3 myLinkedList.get(1); // return 2 myLinkedList.deleteAtIndex(1); // now the linked list is 1->3 myLinkedList.get(1); // return 3

Constraints:

0 <= index, val <= 1000 Please do not use the built-in LinkedList library. At most 2000 calls will be made to get, addAtHead, addAtTail, addAtIndex and deleteAtIndex.

==
Problem Number: 840 URL: https://leetcode.com/problems/minimum-ascii-delete-sum-for-two-strings Title: 712. Minimum ASCII Delete Sum for Two Strings Problem Description: Given two strings s1 and s2, return the lowest ASCII sum of deleted characters to make two strings equal. Example 1: Input: s1 = ”sea”, s2 = ”eat” Output: 231 Explanation: Deleting ”s” from ”sea” adds the ASCII value of ”s” (115) to the sum. Deleting ”t” from ”eat” adds 116 to the sum. At the end, both strings are equal, and 115 + 116 = 231 is the minimum sum possible to achieve this.

Example 2: Input: s1 = ”delete”, s2 = ”leet” Output: 403 Explanation: Deleting ”dee” from ”delete” to turn the string into ”let”, adds 100[d] + 101[e] + 101[e] to the sum. Deleting ”e” from ”leet” adds 101[e] to the sum. At the end, both strings are equal to ”let”, and the answer is 100+101+101+101 = 403. If instead we turned both strings into ”lee” or ”eet”, we would get answers of 433 or 417, which are higher.

Constraints:

1 <= s1.length, s2.length <= 1000 s1 and s2 consist of lowercase English letters.

==
Problem Number: 841 URL: https://leetcode.com/problems/subarray-product-less-than-k Title: 713. Subarray Product Less Than K Problem Description: Given an array of integers nums and an integer k, return the number of contiguous subarrays where the product of all the elements in the subarray is strictly less than k. Example 1: Input: nums = [10,5,2,6], k = 100 Output: 8 Explanation: The 8 subarrays that have product less than 100 are: [10], [5], [2], [6], [10, 5], [5, 2], [2, 6], [5, 2, 6] Note that [10, 5, 2] is not included as the product of 100 is not strictly less than k.

Example 2: Input: nums = [1,2,3], k = 0 Output: 0

Constraints:

1 <= nums.length <= 3 * 104 1 <= nums[i] <= 1000 0 <= k <= 106

==
Problem Number: 842 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-with-transaction-fee Title: 714. Best Time to Buy and Sell Stock with Transaction Fee Problem Description: You are given an array prices where prices[i] is the price of a given stock on the ith day, and an integer fee representing a transaction fee. Find the maximum profit you can achieve. You may complete as many transactions as you like, but you need to pay the transaction fee for each transaction. Note:

You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again). The transaction fee is only charged once for each stock purchase and sale.

Example 1: Input: prices = [1,3,2,8,4,9], fee = 2 Output: 8 Explanation: The maximum profit can be achieved by: - Buying at prices[0] = 1 - Selling at prices[3] = 8 - Buying at prices[4] = 4 - Selling at prices[5] = 9 The total profit is ((8 - 1) - 2) + ((9 - 4) - 2) = 8.

Example 2: Input: prices = [1,3,7,5,10,3], fee = 3 Output: 6

Constraints:

1 <= prices.length <= 5 * 104 1 <= prices[i] < 5 * 104 0 <= fee < 5 * 104

==

Problem Number: 843 URL: https://leetcode.com/problems/maximum-length-of-repeated-subarray Title: 718. Maximum Length of Repeated Subarray Problem Description: Given two integer arrays nums1 and nums2, return the maximum length of a subarray that appears in both arrays. Example 1: Input: nums1 = [1,2,3,2,1], nums2 = [3,2,1,4,7] Output: 3 Explanation: The repeated subarray with maximum length is [3,2,1].

Example 2: Input: nums1 = [0,0,0,0,0], nums2 = [0,0,0,0,0] Output: 5 Explanation: The repeated subarray with maximum length is [0,0,0,0,0].

Constraints:

1 <= nums1.length, nums2.length <= 1000 0 <= nums1[i], nums2[i] <= 100

==

Problem Number: 844 URL: https://leetcode.com/problems/longest-word-in-dictionary Title: 720. Longest Word in Dictionary Problem Description: Given an array of strings words representing an English Dictionary, return the longest word in words that can be built one character at a time by other words in words. If there is more than one possible answer, return the longest word with the smallest lexicographical order. If there is no answer, return the empty string. Note that the word should be built from left to right with each additional character being added to the end of a previous word. Example 1: Input: words = ["w","wo","wor","worl","world"] Output: "world" Explanation: The word "world" can be built one character at a time by "w", "wo", "wor", and "worl".

Example 2: Input: words = ["a","banana","app","appl","ap","apply","apple"] Output: "apple" Explanation: Both "apply" and "apple" can be built from other words in the dictionary. However, "apple" is lexicographically smaller than "apply".

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 30 words[i] consists of lowercase English letters.

==
Problem Number: 845 URL: https://leetcode.com/problems/accounts-merge Title: 721. Accounts Merge Problem Description: Given a list of accounts where each element accounts[i] is a list of strings, where the first element accounts[i][0] is a name, and the rest of the elements are emails representing emails of the account. Now, we would like to merge these accounts. Two accounts definitely belong to the same person if there is some common email to both accounts. Note that even if two accounts have the same name, they may belong to different people as people could have the same name. A person can have any number of accounts initially, but all of their accounts definitely have the same name. After merging the accounts, return the accounts in the following format: the first element of each account is the name, and the rest of the elements are emails in sorted order. The accounts themselves can be returned in any order. Example 1: Input: accounts = [["John","johnsmith@mail.com","john_newyork@mail.com"],["John","johnsmith@mail.com","john00@mail.co
Output: [["John","john00@mail.com","john_newyork@mail.com","johnsmith@mail.com"],["Mary","mary@ma
Explanation: The first and second John's are the same person as they have the common email "johnsmith@mail.com". The third John and Mary are different people as none of their email addresses are used by other accounts. We could return these lists in any order, for example the answer [['Mary', 'mary@mail.com'], ['John', 'johnnybravo@mail.com'], ['John', 'john00@mail.com', 'john_newyork@mail.com', 'johnsmith@mail.com']] would still be accepted.

Example 2: Input: accounts = [["Gabe","Gabe0@m.co","Gabe3@m.co","Gabe1@m.co"],["Kevin","Kevin3@m.c
Output: [["Ethan","Ethan0@m.co","Ethan4@m.co","Ethan5@m.co"],["Gabe","Gabe0@m.co","Gabe1@m.co",

Constraints:

1 <= accounts.length <= 1000 2 <= accounts[i].length <= 10 1 <= accounts[i][j].length <= 30 accounts[i][0] consists of English letters. accounts[i][j] (for j > 0) is a valid email.

==
Problem Number: 846 URL: https://leetcode.com/problems/remove-comments Title: 722. Remove Comments Problem Description: Given a C++ program, remove comments from it. The program source is an array of strings source where source[i] is the ith line of the source code. This represents the result of splitting the original source code string by the newline character '\n'. In C++, there are two types of comments, line comments, and block comments.

The string "//" denotes a line comment, which represents that it and the rest of the characters to the right of it in the same line should be ignored. The string "/*" denotes a block comment, which represents that all characters until the next (non-overlapping) occurrence of "*/" should be ignored. (Here, occurrences happen in reading order: line by line from left to right.) To be clear, the string "/*/" does not yet end the block comment, as the ending would be overlapping the beginning.

The first effective comment takes precedence over others.

For example, if the string "//" occurs in a block comment, it is ignored. Similarly, if the string "/*" occurs in a line or block comment, it is also ignored.

If a certain line of code is empty after removing comments, you must not output that line: each string in the answer list will be non-empty. There will be no control characters, single quote, or double quote characters.

For example, source = "string s = "/* Not a comment. */";" will not be a test case.

Also, nothing else such as defines or macros will interfere with the comments. It is guaranteed that every open block comment will eventually be closed, so "/*" outside of a line or block comment always starts a new comment. Finally, implicit newline characters can be deleted by block comments. Please see the examples below for details. After removing the comments from the source code, return the source code in the same format. Example 1: Input: source = ["/*Test program */", "int main()", "{ ", " // variable declaration ", "int a, b, c;", "/* This is a test", " multiline ", " comment for ", " testing */", "a = b + c;", "}"] Output: ["int main()","{ "," ","int a, b, c;","a = b + c;","}"] Explanation: The line by line code is visualized as below: /*Test program */ int main() { // variable declaration int a, b, c; /* This is a test multiline comment for testing */ a = b + c; } The string /* denotes a block comment, including line 1 and lines 6-9. The string // denotes line 4 as comments. The line by line output code is visualized as below: int main() {

int a, b, c; a = b + c; }

Example 2: Input: source = ["a/*comment", "line", "more_comment*/b"] Output: ["ab"] Explanation: The original source string is "a/*comment\nline\nmore_comment*/b", where we have bolded the newline characters. After deletion, the implicit newline characters are deleted, leaving the string "ab", which when delimited by newline characters becomes ["ab"].

Constraints:

1 <= source.length <= 100 0 <= source[i].length <= 80 source[i] consists of printable ASCII characters. Every open block comment is eventually closed. There are no single-quote or double-quote in the input.

==
Problem Number: 847 URL: https://leetcode.com/problems/split-linked-list-in-parts Title: 725. Split Linked List in Parts Problem Description: Given the head of a singly linked list and an integer k, split the linked list into k consecutive linked list parts. The length of each part should be as equal as possible: no two parts should have a size differing by more than one. This may lead to some parts being null. The parts should be in the order of occurrence in the input list, and parts occurring earlier should always have a size greater than or equal to parts occurring later. Return an array of the k parts. Example 1:

Input: head = [1,2,3], k = 5 Output: [[1],[2],[3],[],[]] Explanation: The first element output[0] has output[0].val = 1, output[0].next = null. The last element output[4] is null, but its string representation as a ListNode is [].

Example 2:

Input: head = [1,2,3,4,5,6,7,8,9,10], k = 3 Output: [[1,2,3,4],[5,6,7],[8,9,10]] Explanation: The input has been split into consecutive parts with size difference at most 1, and earlier parts are a larger size than the later parts.

Constraints:

The number of nodes in the list is in the range [0, 1000]. 0 <= Node.val <= 1000 1 <= k <= 50

==

Problem Number: 848 URL: https://leetcode.com/problems/my-calendar-i Title: 729. My Calendar I Problem Description: You are implementing a program to use as your calendar. We can add a new event if adding the event will not cause a double booking. A double booking happens when two events have some non-empty intersection (i.e., some moment is common to both events.). The event can be represented as a pair of integers start and end that represents a booking on the half-open interval [start, end), the range of real numbers x such that start <= x < end. Implement the MyCalendar class:

MyCalendar() Initializes the calendar object. boolean book(int start, int end) Returns true if the event can be added to the calendar successfully without causing a double booking. Otherwise, return false and do not add the event to the calendar.

Example 1: Input ["MyCalendar", "book", "book", "book"] [[], [10, 20], [15, 25], [20, 30]] Output [null, true, false, true]

Explanation MyCalendar myCalendar = new MyCalendar(); myCalendar.book(10, 20); // return True myCalendar.book(15, 25); // return False, It can not be booked because time 15 is already booked by another event. myCalendar.book(20, 30); // return True, The event can be booked, as the first event takes every time less than 20, but not including 20. Constraints:

0 <= start < end <= 109 At most 1000 calls will be made to book.

==

Problem Number: 849 URL: https://leetcode.com/problems/my-calendar-ii Title: 731. My Calendar II Problem Description: You are implementing a program to use as your calendar. We can add a new event if adding the event will not cause a triple booking. A triple booking happens when three events have some non-empty intersection (i.e., some moment is common to all the three events.). The event can be represented as a pair of integers start and end that represents a booking on the half-open interval [start, end), the range of real numbers x such that start <= x < end. Implement the MyCalendarTwo class:

MyCalendarTwo() Initializes the calendar object. boolean book(int start, int end) Returns true if the event can be added to the calendar successfully without causing a triple booking. Otherwise, return false and do not add the event to the calendar.

Example 1: Input ["MyCalendarTwo", "book", "book", "book", "book", "book", "book"] [[], [10, 20], [50, 60], [10, 40], [5, 15], [5, 10], [25, 55]] Output [null, true, true, true, false, true, true]

Explanation MyCalendarTwo myCalendarTwo = new MyCalendarTwo(); myCalendarTwo.book(10, 20); // return True, The event can be booked. myCalendarTwo.book(50, 60); // return True, The event can be booked. myCalendarTwo.book(10, 40); // return True, The event can be double booked. myCalendarTwo.book(5, 15); // return False, The event cannot be booked, because it would result in a triple booking. myCalendarTwo.book(5, 10); // return True, The event can be booked, as it does not use time 10 which is already double booked. myCalendarTwo.book(25, 55); // return True, The event can be booked, as the time in [25, 40) will be double booked with the third event, the time [40, 50) will be single booked, and the time [50, 55) will be double booked with the second event.

Constraints:

0 <= start < end <= 109 At most 1000 calls will be made to book.

==

Problem Number: 850 URL: https://leetcode.com/problems/asteroid-collision Title: 735. Asteroid Collision Problem Description: We are given an array asteroids of integers representing asteroids in a row. For each asteroid, the absolute value represents its size, and the sign represents its direction (positive meaning right, negative meaning left). Each asteroid moves at the same speed. Find out the state of the asteroids after all collisions. If two asteroids meet, the smaller one will explode. If both are the same size, both will explode. Two asteroids moving in the same direction will never meet. Example 1: Input: asteroids = [5,10,-5] Output: [5,10] Explanation: The 10 and -5 collide resulting in 10. The 5 and 10 never collide.

Example 2: Input: asteroids = [8,-8] Output: [] Explanation: The 8 and -8 collide exploding each other.

Example 3: Input: asteroids = [10,2,-5] Output: [10] Explanation: The 2 and -5 collide resulting in -5. The 10 and -5 collide resulting in 10.

Constraints:

2 <= asteroids.length <= 104 -1000 <= asteroids[i] <= 1000 asteroids[i] != 0

==

Problem Number: 851 URL: https://leetcode.com/problems/monotone-increasing-digits Title: 738. Monotone Increasing Digits Problem Description: An integer has monotone increasing digits if and only if each pair of adjacent digits x and y satisfy x <= y. Given an integer n, return the largest number that is less than or equal to n with monotone increasing digits. Example 1: Input: n = 10 Output: 9

Example 2: Input: n = 1234 Output: 1234

Example 3: Input: n = 332 Output: 299

Constraints:

0 <= n <= 109

==
Problem Number: 852 URL: https://leetcode.com/problems/daily-temperatures Title: 739. Daily Temperatures Problem Description: Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the ith day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead. Example 1: Input: temperatures = [73,74,75,71,69,72,76,73] Output: [1,1,4,2,1,1,0,0] Example 2: Input: temperatures = [30,40,50,60] Output: [1,1,1,0] Example 3: Input: temperatures = [30,60,90] Output: [1,1,0]

Constraints:

1 <= temperatures.length <= 105 30 <= temperatures[i] <= 100

==
Problem Number: 853 URL: https://leetcode.com/problems/delete-and-earn Title: 740. Delete and Earn Problem Description: You are given an integer array nums. You want to maximize the number of points you get by performing the following operation any number of times:

Pick any nums[i] and delete it to earn nums[i] points. Afterwards, you must delete every element equal to nums[i] - 1 and every element equal to nums[i] + 1.

Return the maximum number of points you can earn by applying the above operation some number of times. Example 1: Input: nums = [3,4,2] Output: 6 Explanation: You can perform the following operations: - Delete 4 to earn 4 points. Consequently, 3 is also deleted. nums = [2]. - Delete 2 to earn 2 points. nums = []. You earn a total of 6 points.

Example 2: Input: nums = [2,2,3,3,3,4] Output: 9 Explanation: You can perform the following operations: - Delete a 3 to earn 3 points. All 2's and 4's are also deleted. nums = [3,3]. - Delete a 3 again to earn 3 points. nums = [3]. - Delete a 3 once more to earn 3 points. nums = []. You earn a total of 9 points. Constraints:

1 <= nums.length <= 2 * 104 1 <= nums[i] <= 104

==
Problem Number: 854 URL: https://leetcode.com/problems/network-delay-time Title: 743. Network Delay Time Problem Description: You are given a network of n nodes, labeled from 1 to n. You are also given times, a list of travel times as directed edges times[i] = (ui, vi, wi), where ui is the source node, vi is the target node, and wi is the time it takes for a signal to travel from source to target. We will send a signal from a given node k. Return the minimum time it takes for all the n nodes to receive the signal. If it is impossible for all the n nodes to receive the signal, return -1. Example 1:

Input: times = [[2,1,1],[2,3,1],[3,4,1]], n = 4, k = 2 Output: 2

Example 2: Input: times = [[1,2,1]], n = 2, k = 1 Output: 1

Example 3: Input: times = [[1,2,1]], n = 2, k = 2 Output: -1

Constraints:

1 <= k <= n <= 100 1 <= times.length <= 6000 times[i].length == 3 1 <= ui, vi <= n ui != vi 0 <= wi <= 100 All the pairs (ui, vi) are unique. (i.e., no multiple edges.)

==
Problem Number: 855 URL: https://leetcode.com/problems/open-the-lock Title: 752. Open the Lock Problem Description: You have a lock in front of you with 4 circular wheels. Each wheel has 10 slots: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9'. The wheels can rotate freely and wrap around: for example we can turn '9' to be '0', or '0' to be '9'. Each move consists of turning one wheel one slot. The lock initially starts at '0000', a string representing the state of the 4 wheels. You are given a list of deadends dead ends, meaning if the lock displays any of these codes, the wheels of the lock will stop turning and you will be unable to open it. Given a target representing the value of the wheels that will unlock the lock, return the minimum total number of turns required to open the lock, or -1 if it is impossible. Example 1: Input: deadends = ["0201","0101","0102","1212","2002"], target = "0202" Output: 6 Explanation: A sequence of valid moves would be "0000" -> "1000" -> "1100" -> "1200" -> "1201" -> "1202" -> "0202". Note that a sequence like "0000" -> "0001" -> "0002" -> "0102" -> "0202" would be invalid, because the wheels of the lock become stuck after the display becomes the dead end "0102".

Example 2: Input: deadends = ["8888"], target = "0009" Output: 1 Explanation: We can turn the last wheel in reverse to move from "0000" -> "0009".

Example 3: Input: deadends = ["8887","8889","8878","8898","8788","8988","7888","9888"], target = "8888" Output: -1 Explanation: We cannot reach the target without getting stuck.

Constraints:

1 <= deadends.length <= 500 deadends[i].length == 4 target.length == 4 target will not be in the list deadends. target and deadends[i] consist of digits

only.

== Problem Number: 856 URL: https://leetcode.com/problems/reach-a-number Title: 754. Reach a Number Problem Description: You are standing at position 0 on an infinite number line. There is a destination at position target. You can make some number of moves numMoves so that:

On each move, you can either go left or right. During the ith move (starting from i == 1 to i == numMoves), you take i steps in the chosen direction.

Given the integer target, return the minimum number of moves required (i.e., the minimum numMoves) to reach the destination. Example 1: Input: target = 2 Output: 3 Explanation: On the 1st move, we step from 0 to 1 (1 step). On the 2nd move, we step from 1 to -1 (2 steps). On the 3rd move, we step from -1 to 2 (3 steps).

Example 2: Input: target = 3 Output: 2 Explanation: On the 1st move, we step from 0 to 1 (1 step). On the 2nd move, we step from 1 to 3 (2 steps).

Constraints:

-109 <= target <= 109 target != 0

== Problem Number: 857 URL: https://leetcode.com/problems/pyramid-transition-matrix Title: 756. Pyramid Transition Matrix Problem Description: You are stacking blocks to form a pyramid. Each block has a color, which is represented by a single letter. Each row of blocks contains one less block than the row beneath it and is centered on top. To make the pyramid aesthetically pleasing, there are only specific triangular patterns that are allowed. A triangular pattern consists of a single block stacked on top of two blocks. The patterns are given as a list of three-letter strings allowed, where the first two characters of a pattern represent the left and right bottom blocks respectively, and the third character is the top block.

For example, "ABC" represents a triangular pattern with a 'C' block stacked on top of an 'A' (left) and 'B' (right) block. Note that this is different from "BAC" where 'B' is on the left bottom and 'A' is on the right bottom.

You start with a bottom row of blocks bottom, given as a single string, that you must use as the base of the pyramid. Given bottom and allowed, return true if you can build the pyramid all the way to the top such that every triangular pattern in the pyramid is in allowed, or false otherwise. Example 1:

Input: bottom = "BCD", allowed = ["BCC","CDE","CEA","FFF"] Output: true Explanation: The allowed triangular patterns are shown on the right. Starting from the bottom (level 3), we can build "CE" on level 2 and then build "A" on level 1. There are three triangular patterns in the pyramid, which are "BCC", "CDE", and "CEA". All are allowed.

Example 2:

Input: bottom = "AAAA", allowed = ["AAB","AAC","BCD","BBE","DEF"] Output: false Explanation: The allowed triangular patterns are shown on the right. Starting from the bottom (level 4), there are multiple ways to build level 3, but trying all the possibilites, you will get always stuck before building level 1.

Constraints:

2 <= bottom.length <= 6 0 <= allowed.length <= 216 allowed[i].length == 3 The letters in all input strings are from the set {'A', 'B', 'C', 'D', 'E', 'F'}. All the values of allowed are unique.

==
Problem Number: 858 URL: https://leetcode.com/problems/partition-labels Title: 763. Partition Labels Problem Description: You are given a string s. We want to partition the string into as many parts as possible so that each letter appears in at most one part. Note that the partition is done so that after concatenating all the parts in order, the resultant string should be s. Return a list of integers representing the size of these parts. Example 1: Input: s = "ababcbacadefegdehijhklij" Output: [9,7,8] Explanation: The partition is "ababcbaca", "defegde", "hijhklij". This is a partition so that each letter appears in at most one part. A partition like "ababcbacadefegde", "hijhklij" is incorrect, because it splits s into less parts.

Example 2: Input: s = "eccbbbbdec" Output: [10]

Constraints:

1 <= s.length <= 500 s consists of lowercase English letters.

==
Problem Number: 859 URL: https://leetcode.com/problems/largest-plus-sign Title: 764. Largest Plus Sign Problem Description: You are given an integer n. You have an n x n binary grid grid with all values initially 1's except for some indices given in the array mines. The ith element of the array mines is defined as mines[i] = [xi, yi] where grid[xi][yi] == 0. Return the order of the largest axis-aligned plus sign of 1's contained in grid. If there is none, return 0. An axis-aligned plus sign of 1's of order k has some center grid[r][c] == 1 along with four arms of length k - 1 going up, down, left, and right, and made of 1's. Note that there could be 0's or 1's beyond the arms of the plus sign, only the relevant area of the plus sign is checked for 1's. Example 1:

Input: n = 5, mines = [[4,2]] Output: 2 Explanation: In the above grid, the largest plus sign can only be of order 2. One of them is shown.

Example 2:

Input: n = 1, mines = [[0,0]] Output: 0 Explanation: There is no plus sign, so return 0.

Constraints:

1 <= n <= 500 1 <= mines.length <= 5000 0 <= xi, yi < n All the pairs (xi, yi) are unique.

==
Problem Number: 860 URL: https://leetcode.com/problems/reorganize-string Title: 767. Reorganize String Problem Description: Given a string s, rearrange the characters of s so that any two adjacent characters are not the same. Return any possible rearrangement of s or return "" if not possible. Example 1: Input: s = "aab" Output: "aba" Example 2: Input: s = "aaab" Output: ""

Constraints:

1 <= s.length <= 500 s consists of lowercase English letters.

==
Problem Number: 861 URL: https://leetcode.com/problems/max-chunks-to-make-sorted Title: 769. Max Chunks To Make Sorted Problem Description: You are given an integer array arr of length n that represents a permutation of the integers in the range [0, n - 1]. We split arr into some number of chunks (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array. Return the largest number of chunks we can make to sort the array. Example 1: Input: arr = [4,3,2,1,0] Output: 1 Explanation: Splitting into two or more chunks will not return the required result. For example, splitting into [4, 3], [2, 1, 0] will result in [3, 4, 0, 1, 2], which isn't sorted.

Example 2: Input: arr = [1,0,2,3,4] Output: 4 Explanation: We can split into two chunks, such as [1, 0], [2, 3, 4]. However, splitting into [1, 0], [2], [3], [4] is the highest number of chunks possible.

Constraints:

n == arr.length 1 <= n <= 10 0 <= arr[i] < n All the elements of arr are unique.

==
Problem Number: 862 URL: https://leetcode.com/problems/global-and-local-inversions Title: 775. Global and Local Inversions Problem Description: You are given an integer array nums of length n which represents a permutation of all the integers in the range [0, n - 1]. The number of global inversions is the number of the different pairs (i, j) where:

0 <= i < j < n nums[i] > nums[j]

The number of local inversions is the number of indices i where:

0 <= i < n - 1 nums[i] > nums[i + 1]

Return true if the number of global inversions is equal to the number of local inversions. Example 1: Input: nums = [1,0,2] Output: true Explanation:

There is 1 global inversion and 1 local inversion.

Example 2: Input: nums = [1,2,0] Output: false Explanation: There are 2 global inversions and 1 local inversion.

Constraints:

n == nums.length 1 <= n <= 105 0 <= nums[i] < n All the integers of nums are unique. nums is a permutation of all the numbers in the range [0, n - 1].

==
Problem Number: 863 URL: https://leetcode.com/problems/swap-adjacent-in-lr-string Title: 777. Swap Adjacent in LR String Problem Description: In a string composed of 'L', 'R', and 'X' characters, like "RXXLRXRXL", a move consists of either replacing one occurrence of "XL" with "LX", or replacing one occurrence of "RX" with "XR". Given the starting string start and the ending string end, return True if and only if there exists a sequence of moves to transform one string to the other. Example 1: Input: start = "RXXLRXRXL", end = "XRLXXRRLX" Output: true Explanation: We can transform start to end following these steps: RXXLRXRXL -> XRXLRXRXL -> XRLXRXRXL -> XRLXXRRXL -> XRLXXRRLX

Example 2: Input: start = "X", end = "L" Output: false

Constraints:

1 <= start.length <= 104 start.length == end.length Both start and end will only consist of characters in 'L', 'R', and 'X'.

==
Problem Number: 864 URL: https://leetcode.com/problems/k-th-symbol-in-grammar Title: 779. K-th Symbol in Grammar Problem Description: We build a table of n rows (1-indexed). We start by writing 0 in the 1st row. Now in every subsequent row, we look at the previous row and replace each occurrence of 0 with 01, and each occurrence of 1 with 10.

For example, for n = 3, the 1st row is 0, the 2nd row is 01, and the 3rd row is 0110.

Given two integer n and k, return the kth (1-indexed) symbol in the nth row of a table of n rows. Example 1: Input: n = 1, k = 1 Output: 0 Explanation: row 1: 0

Example 2: Input: n = 2, k = 1 Output: 0 Explanation: row 1: 0 row 2: 01

Example 3: Input: n = 2, k = 2 Output: 1 Explanation: row 1: 0 row 2: 01

Constraints:

1 <= n <= 30 1 <= k <= 2n - 1

==
Problem Number: 865 URL: https://leetcode.com/problems/rabbits-in-forest

Title: 781. Rabbits in Forest Problem Description: There is a forest with an unknown number of rabbits. We asked n rabbits "How many rabbits have the same color as you?" and collected the answers in an integer array answers where answers[i] is the answer of the ith rabbit. Given the array answers, return the minimum number of rabbits that could be in the forest. Example 1: Input: answers = [1,1,2] Output: 5 Explanation: The two rabbits that answered "1" could both be the same color, say red. The rabbit that answered "2" can't be red or the answers would be inconsistent. Say the rabbit that answered "2" was blue. Then there should be 2 other blue rabbits in the forest that didn't answer into the array. The smallest possible number of rabbits in the forest is therefore 5: 3 that answered plus 2 that didn't.

Example 2: Input: answers = [10,10,10] Output: 11

Constraints:

1 <= answers.length <= 1000 0 <= answers[i] < 1000

==
Problem Number: 866 URL: https://leetcode.com/problems/letter-case-permutation Title: 784. Letter Case Permutation Problem Description: Given a string s, you can transform every letter individually to be lowercase or uppercase to create another string. Return a list of all possible strings we could create. Return the output in any order. Example 1: Input: s = "a1b2" Output: ["a1b2","a1B2","A1b2","A1B2"]

Example 2: Input: s = "3z4" Output: ["3z4","3Z4"]

Constraints:

1 <= s.length <= 12 s consists of lowercase English letters, uppercase English letters, and digits.

==
Problem Number: 867 URL: https://leetcode.com/problems/is-graph-bipartite Title: 785. Is Graph Bipartite? Problem Description: There is an undirected graph with n nodes, where each node is numbered between 0 and n - 1. You are given a 2D array graph, where graph[u] is an array of nodes that node u is adjacent to. More formally, for each v in graph[u], there is an undirected edge between node u and node v. The graph has the following properties:

There are no self-edges (graph[u] does not contain u). There are no parallel edges (graph[u] does not contain duplicate values). If v is in graph[u], then u is in graph[v] (the graph is undirected). The graph may not be connected, meaning there may be two nodes u and v such that there is no path between them.

A graph is bipartite if the nodes can be partitioned into two independent sets A and B such that every edge in the graph connects a node in set A and a node in set B. Return true if and only if it is bipartite. Example 1:

Input: graph = [[1,2,3],[0,2],[0,1,3],[0,2]] Output: false Explanation: There is no way to partition the nodes into two independent sets such that every edge connects a node in one and a node in the other. Example 2:

Input: graph = [[1,3],[0,2],[1,3],[0,2]] Output: true Explanation: We can partition the nodes into two sets: {0, 2} and {1, 3}. Constraints:

graph.length == n 1 <= n <= 100 0 <= graph[u].length < n 0 <= graph[u][i] <= n - 1 graph[u] does not contain u. All the values of graph[u] are unique. If graph[u] contains v, then graph[v] contains u.

==

Problem Number: 868 URL: https://leetcode.com/problems/k-th-smallest-prime-fraction Title: 786. K-th Smallest Prime Fraction Problem Description: You are given a sorted integer array arr containing 1 and prime numbers, where all the integers of arr are unique. You are also given an integer k. For every i and j where 0 <= i < j < arr.length, we consider the fraction arr[i] / arr[j]. Return the kth smallest fraction considered. Return your answer as an array of integers of size 2, where answer[0] == arr[i] and answer[1] == arr[j]. Example 1: Input: arr = [1,2,3,5], k = 3 Output: [2,5] Explanation: The fractions to be considered in sorted order are: 1/5, 1/3, 2/5, 1/2, 3/5, and 2/3. The third fraction is 2/5.

Example 2: Input: arr = [1,7], k = 1 Output: [1,7]

Constraints:

2 <= arr.length <= 1000 1 <= arr[i] <= 3 * 104 arr[0] == 1 arr[i] is a prime number for i > 0. All the numbers of arr are unique and sorted in strictly increasing order. 1 <= k <= arr.length * (arr.length - 1) / 2

Follow up: Can you solve the problem with better than O(n2) complexity?

==

Problem Number: 869 URL: https://leetcode.com/problems/cheapest-flights-within-k-stops Title: 787. Cheapest Flights Within K Stops Problem Description: There are n cities connected by some number of flights. You are given an array flights where flights[i] = [fromi, toi, pricei] indicates that there is a flight from city fromi to city toi with cost pricei. You are also given three integers src, dst, and k, return the cheapest price from src to dst with at most k stops. If there is no such route, return -1. Example 1:

Input: n = 4, flights = [[0,1,100],[1,2,100],[2,0,100],[1,3,600],[2,3,200]], src = 0, dst = 3, k = 1 Output: 700 Explanation: The graph is shown above. The optimal path with at most 1 stop from city 0 to 3 is marked in red and has cost 100 + 600 = 700. Note that the path through cities [0,1,2,3] is cheaper but is invalid because it uses 2 stops.

Example 2:

Input: n = 3, flights = [[0,1,100],[1,2,100],[0,2,500]], src = 0, dst = 2, k = 1 Output: 200 Explanation: The graph is shown above. The optimal path with at most 1 stop from city 0 to 2 is marked in red and has cost 100 + 100 = 200.

Example 3:

Input: n = 3, flights = [[0,1,100],[1,2,100],[0,2,500]], src = 0, dst = 2, k = 0 Output: 500 Explanation: The graph is shown above. The optimal path with no stops from city 0 to 2 is marked in red and has cost 500.

Constraints:

1 <= n <= 100 0 <= flights.length <= (n * (n - 1) / 2) flights[i].length == 3 0 <= fromi, toi < n fromi != toi 1 <= pricei <= 104 There will not be any multiple flights between two cities. 0 <= src, dst, k < n src != dst

==
Problem Number: 870 URL: https://leetcode.com/problems/rotated-digits Title: 788. Rotated Digits Problem Description: An integer x is a good if after rotating each digit individually by 180 degrees, we get a valid number that is different from x. Each digit must be rotated - we cannot choose to leave it alone. A number is valid if each digit remains a digit after rotation. For example:

0, 1, and 8 rotate to themselves, 2 and 5 rotate to each other (in this case they are rotated in a different direction, in other words, 2 or 5 gets mirrored), 6 and 9 rotate to each other, and the rest of the numbers do not rotate to any other number and become invalid.

Given an integer n, return the number of good integers in the range [1, n]. Example 1: Input: n = 10 Output: 4 Explanation: There are four good numbers in the range [1, 10] : 2, 5, 6, 9. Note that 1 and 10 are not good numbers, since they remain unchanged after rotating.

Example 2: Input: n = 1 Output: 0

Example 3: Input: n = 2 Output: 1

Constraints:

1 <= n <= 104

==
Problem Number: 871 URL: https://leetcode.com/problems/escape-the-ghosts Title: 789. Escape The Ghosts Problem Description: You are playing a simplified PAC-MAN game on an infinite 2-D grid. You start at the point [0, 0], and you are given a destination point target = [xtarget, ytarget] that you are trying to get to. There are several ghosts on the map with their starting positions given as a 2D array ghosts, where ghosts[i] = [xi, yi] represents the starting position of the ith ghost. All inputs are integral coordinates. Each turn, you and all the ghosts may independently choose to either move 1 unit in any of the four cardinal directions: north, east, south, or west, or stay still. All actions happen simultaneously. You escape if and only if you can reach the target before any ghost reaches you. If you reach any square (including the

target) at the same time as a ghost, it does not count as an escape. Return true if it is possible to escape regardless of how the ghosts move, otherwise return false. Example 1: Input: ghosts = [[1,0],[0,3]], target = [0,1] Output: true Explanation: You can reach the destination (0, 1) after 1 turn, while the ghosts located at (1, 0) and (0, 3) cannot catch up with you.

Example 2: Input: ghosts = [[1,0]], target = [2,0] Output: false Explanation: You need to reach the destination (2, 0), but the ghost at (1, 0) lies between you and the destination.

Example 3: Input: ghosts = [[2,0]], target = [1,0] Output: false Explanation: The ghost can reach the target at the same time as you.

Constraints:

1 <= ghosts.length <= 100 ghosts[i].length == 2 -104 <= xi, yi <= 104 There can be multiple ghosts in the same location. target.length == 2 -104 <= xtarget, ytarget <= 104

==

Problem Number: 872 URL: https://leetcode.com/problems/domino-and-tromino-tiling Title: 790. Domino and Tromino Tiling Problem Description: You have two types of tiles: a 2 x 1 domino shape and a tromino shape. You may rotate these shapes.

Given an integer n, return the number of ways to tile an 2 x n board. Since the answer may be very large, return it modulo 109 + 7. In a tiling, every square must be covered by a tile. Two tilings are different if and only if there are two 4-directionally adjacent cells on the board such that exactly one of the tilings has both squares occupied by a tile. Example 1:

Input: n = 3 Output: 5 Explanation: The five different ways are show above.

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 1000

==

Problem Number: 873 URL: https://leetcode.com/problems/custom-sort-string Title: 791. Custom Sort String Problem Description: You are given two strings order and s. All the characters of order are unique and were sorted in some custom order previously. Permute the characters of s so that they match the order that order was sorted. More specifically, if a character x occurs before a character y in order, then x should occur before y in the permuted string. Return any permutation of s that satisfies this property. Example 1: Input: order = "cba", s = "abcd" Output: "cbad" Explanation: "a", "b", "c" appear in order, so the order of "a", "b", "c" should be "c", "b", and "a". Since "d" does not appear in order, it can be at any position in the returned string. "dcba", "cdba", "cbda" are also valid outputs.

Example 2: Input: order = ”cbafg”, s = ”abcd” Output: ”cbad”

Constraints:

1 <= order.length <= 26 1 <= s.length <= 200 order and s consist of lowercase English letters. All the characters of order are unique.

==
Problem Number: 874 URL: https://leetcode.com/problems/number-of-matching-subsequences Title: 792. Number of Matching Subsequences Problem Description: Given a string s and an array of strings words, return the number of words[i] that is a subsequence of s. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, ”ace” is a subsequence of ”abcde”.

Example 1: Input: s = ”abcde”, words = [”a”,”bb”,”acd”,”ace”] Output: 3 Explanation: There are three strings in words that are a subsequence of s: ”a”, ”acd”, ”ace”.

Example 2: Input: s = ”dsahjpjauf”, words = [”ahjpjau”,”ja”,”ahbwzgqnuk”,”tnmlanowax”] Output: 2

Constraints:

1 <= s.length <= 5 * 104 1 <= words.length <= 5000 1 <= words[i].length <= 50 s and words[i] consist of only lowercase English letters.

==
Problem Number: 875 URL: https://leetcode.com/problems/valid-tic-tac-toe-state Title: 794. Valid Tic-Tac-Toe State Problem Description: Given a Tic-Tac-Toe board as a string array board, return true if and only if it is possible to reach this board position during the course of a valid tic-tac-toe game. The board is a 3 x 3 array that consists of characters ’ ’, ’X’, and ’O’. The ’ ’ character represents an empty square. Here are the rules of Tic-Tac-Toe:

Players take turns placing characters into empty squares ’ ’. The first player always places ’X’ characters, while the second player always places ’O’ characters. ’X’ and ’O’ characters are always placed into empty squares, never filled ones. The game ends when there are three of the same (non-empty) character filling any row, column, or diagonal. The game also ends if all squares are non-empty. No more moves can be played if the game is over.

Example 1:

Input: board = [”O ”,” ”,” ”] Output: false Explanation: The first player always plays ”X”.

Example 2:

Input: board = [”XOX”,” X ”,” ”] Output: false Explanation: Players take turns making moves.

Example 3:

Input: board = ["XOX","O O","XOX"] Output: true

Constraints:

board.length == 3 board[i].length == 3 board[i][j] is either 'X', 'O', or ' '.

==

Problem Number: 876 URL: https://leetcode.com/problems/number-of-subarrays-with-bounded-maximum Title: 795. Number of Subarrays with Bounded Maximum Problem Description: Given an integer array nums and two integers left and right, return the number of contiguous non-empty subarrays such that the value of the maximum array element in that subarray is in the range [left, right]. The test cases are generated so that the answer will fit in a 32-bit integer. Example 1: Input: nums = [2,1,4,3], left = 2, right = 3 Output: 3 Explanation: There are three subarrays that meet the requirements: [2], [2, 1], [3].

Example 2: Input: nums = [2,9,2,5,6], left = 2, right = 8 Output: 7

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109 0 <= left <= right <= 109

==

Problem Number: 877 URL: https://leetcode.com/problems/all-paths-from-source-to-target Title: 797. All Paths From Source to Target Problem Description: Given a directed acyclic graph (DAG) of n nodes labeled from 0 to n - 1, find all possible paths from node 0 to node n - 1 and return them in any order. The graph is given as follows: graph[i] is a list of all nodes you can visit from node i (i.e., there is a directed edge from node i to node graph[i][j]). Example 1:

Input: graph = [[1,2],[3],[3],[]] Output: [[0,1,3],[0,2,3]] Explanation: There are two paths: 0 -> 1 -> 3 and 0 -> 2 -> 3.

Example 2:

Input: graph = [[4,3,1],[3,2,4],[3],[4],[]] Output: [[0,4],[0,3,4],[0,1,3,4],[0,1,2,3,4],[0,1,4]]

Constraints:

n == graph.length 2 <= n <= 15 0 <= graph[i][j] < n graph[i][j] != i (i.e., there will be no self-loops). All the elements of graph[i] are unique. The input graph is guaranteed to be a DAG.

==

Problem Number: 878 URL: https://leetcode.com/problems/champagne-tower Title: 799. Champagne Tower Problem Description: We stack glasses in a pyramid, where the first row has 1 glass, the second row has 2 glasses, and so on until the 100th row. Each glass holds one cup of champagne. Then, some champagne is poured into the first glass at the top. When the topmost glass is full, any excess liquid poured will fall equally to the glass immediately to the left and right of it. When those glasses become full, any excess champagne will fall equally to the left and right of those glasses, and so on. (A glass at the bottom row has its excess champagne fall on the floor.) For example, after one cup of champagne is poured, the top most glass is full. After two cups of champagne are poured, the two glasses on the second row are half full. After three cups of champagne are poured, those two cups become full - there are 3 full glasses total now. After four cups of champagne are poured, the third row has the middle glass half full, and the two outside glasses are a quarter full, as pictured below.

Now after pouring some non-negative integer cups of champagne, return how full the jth glass in the ith row is (both i and j are 0-indexed.) Example 1: Input: poured = 1, query_row = 1, query_glass = 1 Output: 0.00000 Explanation: We poured 1 cup of champange to the top glass of the tower (which is indexed as (0, 0)). There will be no excess liquid so all the glasses under the top glass will remain empty.

Example 2: Input: poured = 2, query_row = 1, query_glass = 1 Output: 0.50000 Explanation: We poured 2 cups of champange to the top glass of the tower (which is indexed as (0, 0)). There is one cup of excess liquid. The glass indexed as (1, 0) and the glass indexed as (1, 1) will share the excess liquid equally, and each will get half cup of champange.

Example 3: Input: poured = 100000009, query_row = 33, query_glass = 17 Output: 1.00000

Constraints:

0 <= poured <= 109 0 <= query_glass <= query_row < 100

==
Problem Number: 879 URL: https://leetcode.com/problems/find-eventual-safe-states Title: 802. Find Eventual Safe States Problem Description: There is a directed graph of n nodes with each node labeled from 0 to n - 1. The graph is represented by a 0-indexed 2D integer array graph where graph[i] is an integer array of nodes adjacent to node i, meaning there is an edge from node i to each node in graph[i]. A node is a terminal node if there are no outgoing edges. A node is a safe node if every possible path starting from that node leads to a terminal node (or another safe node). Return an array containing all the safe nodes of the graph. The answer should be sorted in ascending order. Example 1:

Input: graph = [[1,2],[2,3],[5],[0],[5],[],[]] Output: [2,4,5,6] Explanation: The given graph is shown above. Nodes 5 and 6 are terminal nodes as there are no outgoing edges from either of them. Every path starting at nodes 2, 4, 5, and 6 all lead to either node 5 or 6. Example 2: Input: graph = [[1,2,3,4],[1,2],[3,4],[0,4],[]] Output: [4] Explanation: Only node 4 is a terminal node, and every path starting at node 4 leads to node 4.

Constraints:

n == graph.length 1 <= n <= 104 0 <= graph[i].length <= n 0 <= graph[i][j] <= n - 1 graph[i] is sorted in a strictly increasing order. The graph may contain self-loops. The number of edges in the graph will be in the range [1, 4 * 104].

==
Problem Number: 880 URL: https://leetcode.com/problems/max-increase-to-keep-city-skyline Title: 807. Max Increase to Keep City Skyline Problem Description: There is a city composed of n x n blocks, where each block contains a single building shaped like a vertical square prism. You are given a 0-indexed n x n integer matrix grid where grid[r][c] represents the height of the building located in the block at row r and column c. A city's skyline is the outer contour formed by all the building when viewing the side of the city from a distance. The skyline from each cardinal direction north, east, south, and west may be different. We are allowed to increase the height of any number of buildings by any amount (the amount can be different per building). The height of a 0-height building can also be increased. However, increasing the height of a building should not affect the city's skyline from any cardinal direction. Return the maximum total sum that the height of the buildings can be increased by without changing the city's skyline from any cardinal direction. Example 1:

Input: grid = [[3,0,8,4],[2,4,5,7],[9,2,6,3],[0,3,1,0]] Output: 35 Explanation: The building heights are shown in the center of the above image. The skylines when viewed from each cardinal direction are drawn in red. The grid after increasing the height of buildings without affecting skylines is: gridNew = [[8, 4, 8, 7], [7, 4, 7, 7], [9, 4, 8, 7], [3, 3, 3, 3]]

Example 2: Input: grid = [[0,0,0],[0,0,0],[0,0,0]] Output: 0 Explanation: Increasing the height of any building will result in the skyline changing.

Constraints:

n == grid.length n == grid[r].length 2 <= n <= 50 0 <= grid[r][c] <= 100

==
Problem Number: 881 URL: https://leetcode.com/problems/soup-servings Title: 808. Soup Servings Problem Description: There are two types of soup: type A and type B. Initially, we have n ml of each type of soup. There are four kinds of operations:

Serve 100 ml of soup A and 0 ml of soup B, Serve 75 ml of soup A and 25 ml of soup B, Serve 50 ml of soup A and 50 ml of soup B, and Serve 25 ml of soup A and 75 ml of soup B.

When we serve some soup, we give it to someone, and we no longer have it. Each turn, we will choose from the four operations with an equal probability 0.25. If the remaining volume of soup is not enough to complete the operation, we will serve as much as possible. We stop once we no longer have some quantity of both types of soup. Note that we do not have an operation where all 100 ml's of soup B are used first. Return the probability that soup A will be empty first, plus half the probability that A and B become empty at the same time. Answers within 10-5 of the actual answer will be accepted.   Example 1: Input: n = 50 Output: 0.62500 Explanation: If we choose the first two operations, A will become empty first. For the third operation, A and B will become empty at the same time. For the fourth operation, B will become empty first. So the total probability of A becoming empty first plus half the probability that A and B become empty at the same time, is 0.25 * (1 + 1 + 0.5 + 0) = 0.625.

Example 2: Input: n = 100 Output: 0.71875

Constraints:

0 <= n <= 109

==
Problem Number: 882 URL: https://leetcode.com/problems/expressive-words Title: 809. Expressive Words Problem Description: Sometimes people repeat letters to represent extra feeling. For example:

"hello" -> "heeellooo" "hi" -> "hiiii"

In these strings like "heeellooo", we have groups of adjacent letters that are all the same: "h", "eee", "ll", "ooo". You are given a string s and an array of query strings words. A query word is stretchy if it can be made to be equal to s by any number of applications of the following extension operation: choose a group consisting of characters c, and add some number of characters c to the group so that the size of the group is three or more.

For example, starting with "hello", we could do an extension on the group "o" to get "hellooo", but we cannot get "helloo" since the group "oo" has a size less than three. Also, we could do another extension like "ll" -> "lllll" to get "helllllooo". If s = "helllllooo", then the query word "hello" would be stretchy because of these two extension operations: query = "hello" -> "hellooo" -> "helllllooo" = s.

Return the number of query strings that are stretchy.   Example 1: Input: s = "heeellooo", words = ["hello", "hi", "helo"] Output: 1 Explanation: We can extend "e" and "o" in the word "hello" to get "heeellooo". We can't extend "helo" to get "heeellooo" because the group "ll" is not size 3 or more.

Example 2: Input: s = "zzzzzyyyyy", words = ["zzyy","zy","zyy"] Output: 3

Constraints:

1 <= s.length, words.length <= 100 1 <= words[i].length <= 100 s and words[i] consist of lowercase letters.

==
Problem Number: 883 URL: https://leetcode.com/problems/subdomain-visit-count Title: 811. Subdomain Visit Count Problem Description: A website

domain ”discuss.leetcode.com” consists of various subdomains. At the top level, we have ”com”, at the next level, we have ”leetcode.com” and at the lowest level, ”discuss.leetcode.com”. When we visit a domain like ”discuss.leetcode.com”, we will also visit the parent domains ”leetcode.com” and ”com” implicitly. A count-paired domain is a domain that has one of the two formats ”rep d1.d2.d3” or ”rep d1.d2” where rep is the number of visits to the domain and d1.d2.d3 is the domain itself.

For example, ”9001 discuss.leetcode.com” is a count-paired domain that indicates that discuss.leetcode.com was visited 9001 times.

Given an array of count-paired domains cpdomains, return an array of the count-paired domains of each subdomain in the input. You may return the answer in any order. Example 1: Input: cpdomains = [”9001 discuss.leetcode.com”] Output: [”9001 leetcode.com”,”9001 discuss.leetcode.com”,”9001 com”] Explanation: We only have one website domain: ”discuss.leetcode.com”. As discussed above, the subdomain ”leetcode.com” and ”com” will also be visited. So they will all be visited 9001 times.

Example 2: Input: cpdomains = [”900 google.mail.com”, ”50 yahoo.com”, ”1 intel.mail.com”, ”5 wiki.org”] Output: [”901 mail.com”,”50 yahoo.com”,”900 google.mail.com”,”5 wiki.org”,”5 org”,”1 intel.mail.com”,”951 com”] Explanation: We will visit ”google.mail.com” 900 times, ”yahoo.com” 50 times, ”intel.mail.com” once and ”wiki.org” 5 times. For the subdomains, we will visit ”mail.com” 900 + 1 = 901 times, ”com” 900 + 50 + 1 = 951 times, and ”org” 5 times.

Constraints:

1 <= cpdomain.length <= 100 1 <= cpdomain[i].length <= 100 cpdomain[i] follows either the ”repi d1i.d2i.d3i” format or the ”repi d1i.d2i” format. repi is an integer in the range [1, 104]. d1i, d2i, and d3i consist of lowercase English letters.

==

Problem Number: 884 URL: https://leetcode.com/problems/largest-sum-of-averages Title: 813. Largest Sum of Averages Problem Description: You are given an integer array nums and an integer k. You can partition the array into at most k non-empty adjacent subarrays. The score of a partition is the sum of the averages of each subarray. Note that the partition must use every integer in nums, and that the score is not necessarily an integer. Return the maximum score you can achieve of all the possible partitions. Answers within 10-6 of the actual answer will be accepted. Example 1: Input: nums = [9,1,2,3,9], k = 3 Output: 20.00000 Explanation: The best choice is to partition nums into [9], [1, 2, 3], [9]. The answer is 9 + (1 + 2 + 3) / 3 + 9 = 20. We could have also partitioned nums into [9, 1], [2], [3, 9], for example. That partition would lead to a score of 5 + 2 + 6 = 13, which is worse.

Example 2: Input: nums = [1,2,3,4,5,6,7], k = 4 Output: 20.50000

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 104 1 <= k <= nums.length

==
Problem Number: 885 URL: https://leetcode.com/problems/binary-tree-pruning Title: 814. Binary Tree Pruning Problem Description: Given the root of a binary tree, return the same tree where every subtree (of the given tree) not containing a 1 has been removed. A subtree of a node node is node plus every node that is a descendant of node. Example 1:

Input: root = [1,null,0,0,1] Output: [1,null,0,null,1] Explanation: Only the red nodes satisfy the property "every subtree not containing a 1". The diagram on the right represents the answer.

Example 2:

Input: root = [1,0,1,0,0,0,1] Output: [1,null,1,null,1]

Example 3:

Input: root = [1,1,0,1,1,0,1,0] Output: [1,1,0,1,1,null,1]

Constraints:

The number of nodes in the tree is in the range [1, 200]. Node.val is either 0 or 1.

==
Problem Number: 886 URL: https://leetcode.com/problems/ambiguous-coordinates Title: 816. Ambiguous Coordinates Problem Description: We had some 2-dimensional coordinates, like "(1, 3)" or "(2, 0.5)". Then, we removed all commas, decimal points, and spaces and ended up with the string s.

For example, "(1, 3)" becomes s = "(13)" and "(2, 0.5)" becomes s = "(205)".

Return a list of strings representing all possibilities for what our original coordinates could have been. Our original representation never had extraneous zeroes, so we never started with numbers like "00", "0.0", "0.00", "1.0", "001", "00.01", or any other number that can be represented with fewer digits. Also, a decimal point within a number never occurs without at least one digit occurring before it, so we never started with numbers like ".1". The final answer list can be returned in any order. All coordinates in the final answer have exactly one space between them (occurring after the comma.) Example 1: Input: s = "(123)" Output: ["(1, 2.3)","(1, 23)","(1.2, 3)","(12, 3)"]

Example 2: Input: s = "(0123)" Output: ["(0, 1.23)","(0, 12.3)","(0, 123)","(0.1, 2.3)","(0.1, 23)","(0.12, 3)"] Explanation: 0.0, 00, 0001 or 00.01 are not allowed.

Example 3: Input: s = "(00011)" Output: ["(0, 0.011)","(0.001, 1)"]

Constraints:

4 <= s.length <= 12 s[0] == '(' and s[s.length - 1] == ')'. The rest of s are digits.

==
Problem Number: 887 URL: https://leetcode.com/problems/linked-list-components Title: 817. Linked List Components Problem Description: You are given the head of a linked list containing unique integer values and an integer array nums that is a subset of the linked list values. Return the number of connected components in nums where two values are connected if they appear consecutively in the linked list. Example 1:

Input: head = [0,1,2,3], nums = [0,1,3] Output: 2 Explanation: 0 and 1 are connected, so [0, 1] and [3] are the two connected components.

Example 2:

Input: head = [0,1,2,3,4], nums = [0,3,1,4] Output: 2 Explanation: 0 and 1 are connected, 3 and 4 are connected, so [0, 1] and [3, 4] are the two connected components.

Constraints:

The number of nodes in the linked list is n. 1 <= n <= 104 0 <= Node.val < n All the values Node.val are unique. 1 <= nums.length <= n 0 <= nums[i] < n All the values of nums are unique.

==
Problem Number: 888 URL: https://leetcode.com/problems/short-encoding-of-words Title: 820. Short Encoding of Words Problem Description: A valid encoding of an array of words is any reference string s and array of indices indices such that:

words.length == indices.length The reference string s ends with the '#' character. For each index indices[i], the substring of s starting from indices[i] and up to (but not including) the next '#' character is equal to words[i].

Given an array of words, return the length of the shortest reference string s possible of any valid encoding of words. Example 1: Input: words = ["time", "me", "bell"] Output: 10 Explanation: A valid encoding would be s = "time#bell#" and indices = [0, 2, 5]. words[0] = "time", the substring of s starting from indices[0] = 0 to the next '#' is underlined in "time#bell#" words[1] = "me", the substring of s starting from indices[1] = 2 to the next '#' is underlined in "time#bell#" words[2] = "bell", the substring of s starting from indices[2] = 5 to the next '#' is underlined in "time#bell#"

Example 2: Input: words = ["t"] Output: 2 Explanation: A valid encoding would be s = "t#" and indices = [0].

Constraints:

1 <= words.length <= 2000 1 <= words[i].length <= 7 words[i] consists of only lowercase letters.

==
Problem Number: 889 URL: https://leetcode.com/problems/card-flipping-game Title: 822. Card Flipping Game Problem Description: You are given two 0-indexed integer arrays fronts and backs of length n, where the ith card has the positive integer fronts[i] printed on the front and backs[i] printed on the back. Initially, each card is placed on a table such that the front number is facing up and the other is facing down. You may flip over any number of cards (possibly zero). After flipping the cards, an integer is considered good if it is facing down on some card and not facing up on any card. Return the minimum possible good integer after flipping the cards. If there are no good integers, return 0. Example 1: Input: fronts = [1,2,4,4,7], backs = [1,3,4,1,3] Output: 2 Explanation: If we flip the second card, the face up numbers are [1,3,4,4,7] and the face down are [1,2,4,1,3]. 2 is the minimum good integer as it appears facing down but not facing up. It can be shown that 2 is the minimum possible good integer obtainable after flipping some cards.

Example 2: Input: fronts = [1], backs = [1] Output: 0 Explanation: There are no good integers no matter how we flip the cards, so we return 0.

Constraints:

n == fronts.length == backs.length 1 <= n <= 1000 1 <= fronts[i], backs[i] <= 2000

==
Problem Number: 890 URL: https://leetcode.com/problems/binary-trees-with-factors Title: 823. Binary Trees With Factors Problem Description: Given an array of unique integers, arr, where each integer arr[i] is strictly greater than 1. We make a binary tree using these integers, and each number may be used for any number of times. Each non-leaf node's value should be equal to the product of the values of its children. Return the number of binary trees we can make. The answer may be too large so return the answer modulo 109 + 7. Example 1: Input: arr = [2,4] Output: 3 Explanation: We can make these trees: [2], [4], [4, 2, 2] Example 2: Input: arr = [2,4,5,10] Output: 7 Explanation: We can make these trees: [2], [4], [5], [10], [4, 2, 2], [10, 2, 5], [10, 5, 2]. Constraints:

1 <= arr.length <= 1000 2 <= arr[i] <= 109 All the values of arr are unique.

==
Problem Number: 891 URL: https://leetcode.com/problems/friends-of-appropriate-ages Title: 825. Friends Of Appropriate Ages Problem Description: There are n persons on a social media website. You are given an integer array ages where ages[i] is the age of the ith person. A Person x will not send a friend request to a person y (x != y) if any of the following conditions is true:

age[y] <= 0.5 * age[x] + 7 age[y] > age[x] age[y] > 100 && age[x] < 100

Otherwise, x will send a friend request to y. Note that if x sends a request to y, y will not necessarily send a request to x. Also, a person will not send a friend request to themself. Return the total number of friend requests made. Example 1: Input: ages = [16,16] Output: 2 Explanation: 2 people friend request each other.

Example 2: Input: ages = [16,17,18] Output: 2 Explanation: Friend requests are made 17 -> 16, 18 -> 17.

Example 3: Input: ages = [20,30,100,110,120] Output: 3 Explanation: Friend requests are made 110 -> 100, 120 -> 110, 120 -> 100.

Constraints:

n == ages.length 1 <= n <= 2 * 104 1 <= ages[i] <= 120

==
Problem Number: 892 URL: https://leetcode.com/problems/most-profit-assigning-work Title: 826. Most Profit Assigning Work Problem Description: You have n jobs and m workers. You are given three arrays: difficulty, profit, and worker where:

difficulty[i] and profit[i] are the difficulty and the profit of the ith job, and worker[j] is the ability of jth worker (i.e., the jth worker can only complete a job with difficulty at most worker[j]).

Every worker can be assigned at most one job, but one job can be completed multiple times.

For example, if three workers attempt the same job that pays $1, then the total profit will be $3. If a worker cannot complete any job, their profit is $0.

Return the maximum profit we can achieve after assigning the workers to the jobs. Example 1: Input: difficulty = [2,4,6,8,10], profit = [10,20,30,40,50], worker = [4,5,6,7] Output: 100 Explanation: Workers are assigned jobs of difficulty [4,4,6,6] and they get a profit of [20,20,30,30] separately.

Example 2: Input: difficulty = [85,47,57], profit = [24,66,99], worker = [40,25,25] Output: 0

Constraints:

n == difficulty.length n == profit.length m == worker.length 1 <= n, m <= 104 1 <= difficulty[i], profit[i], worker[i] <= 105

==
Problem Number: 893 URL: https://leetcode.com/problems/masking-personal-information Title: 831. Masking Personal Information Problem Description: You are given a personal information string s, representing either an email address or a phone number. Return the masked personal information using the below rules. Email address: An email address is:

A name consisting of uppercase and lowercase English letters, followed by The '@' symbol, followed by The domain consisting of uppercase and lowercase English letters with a dot '.' somewhere in the middle (not the first or last character).

To mask an email:

The uppercase letters in the name and domain must be converted to lowercase letters. The middle letters of the name (i.e., all but the first and last letters) must be replaced by 5 asterisks "*****".

Phone number: A phone number is formatted as follows:

The phone number contains 10-13 digits. The last 10 digits make up the local number. The remaining 0-3 digits, in the beginning, make up the country code. Separation characters from the set {'+', '-', '(', ')', ' '} separate the above digits in some way.

To mask a phone number:

Remove all separation characters. The masked phone number should have the form:

"***-***-XXXX" if the country code has 0 digits. "+*-***-***-XXXX" if the country code has 1 digit. "+**-***-***-XXXX" if the country code has 2 digits. "+***-***-***-XXXX" if the country code has 3 digits.

"XXXX" is the last 4 digits of the local number.

Example 1: Input: s = "LeetCode@LeetCode.com" Output: "l*****e@leetcode.com" Explanation: s is an email address. The name and domain are converted to lowercase, and the middle of the name is replaced by 5 asterisks.

Example 2: Input: s = "AB@qq.com" Output: "a*****b@qq.com" Explanation: s is an email address. The name and domain are converted to lowercase, and the middle of the name is replaced by 5 asterisks. Note that even though "ab" is 2 characters, it still must have 5 asterisks in the middle.

Example 3: Input: s = "1(234)567-890" Output: "***-***-7890" Explanation: s is a phone number. There are 10 digits, so the local number is 10 digits and the country code is 0 digits. Thus, the resulting masked number is "***-***-7890".

Constraints:

s is either a valid email or a phone number. If s is an email:

8 <= s.length <= 40 s consists of uppercase and lowercase English letters and exactly one '@' symbol and '.' symbol.

If s is a phone number:

10 <= s.length <= 20 s consists of digits, spaces, and the symbols '(', ')', '-', and '+'.

==
Problem Number: 894 URL: https://leetcode.com/problems/find-and-replace-

in-string Title: 833. Find And Replace in String Problem Description: You are given a 0-indexed string s that you must perform k replacement operations on. The replacement operations are given as three 0-indexed parallel arrays, indices, sources, and targets, all of length k. To complete the ith replacement operation:

Check if the substring sources[i] occurs at index indices[i] in the original string s. If it does not occur, do nothing. Otherwise if it does occur, replace that substring with targets[i].

For example, if s = "abcd", indices[i] = 0, sources[i] = "ab", and targets[i] = "eee", then the result of this replacement will be "eeecd". All replacement operations must occur simultaneously, meaning the replacement operations should not affect the indexing of each other. The testcases will be generated such that the replacements will not overlap.

For example, a testcase with s = "abc", indices = [0, 1], and sources = ["ab","bc"] will not be generated because the "ab" and "bc" replacements overlap.

Return the resulting string after performing all replacement operations on s. A substring is a contiguous sequence of characters in a string. Example 1:

Input: s = "abcd", indices = [0, 2], sources = ["a", "cd"], targets = ["eee", "ffff"] Output: "eeebffff" Explanation: "a" occurs at index 0 in s, so we replace it with "eee". "cd" occurs at index 2 in s, so we replace it with "ffff".

Example 2:

Input: s = "abcd", indices = [0, 2], sources = ["ab","ec"], targets = ["eee","ffff"] Output: "eeecd" Explanation: "ab" occurs at index 0 in s, so we replace it with "eee". "ec" does not occur at index 2 in s, so we do nothing.

Constraints:

1 <= s.length <= 1000 k == indices.length == sources.length == targets.length 1 <= k <= 100 0 <= indexes[i] < s.length 1 <= sources[i].length, targets[i].length <= 50 s consists of only lowercase English letters. sources[i] and targets[i] consist of only lowercase English letters.

==
Problem Number: 895 URL: https://leetcode.com/problems/image-overlap Title: 835. Image Overlap Problem Description: You are given two images, img1 and img2, represented as binary, square matrices of size n x n. A binary matrix has only 0s and 1s as values. We translate one image however we choose by sliding all the 1 bits left, right, up, and/or down any number of units. We then place it on top of the other image. We can then calculate the overlap by counting the number of positions that have a 1 in both images. Note also that a translation does not include any kind of rotation. Any 1 bits that are translated outside of the matrix borders are erased. Return the largest possible overlap. Example 1:

Input: img1 = [[1,1,0],[0,1,0],[0,1,0]], img2 = [[0,0,0],[0,1,1],[0,0,1]] Output: 3 Explanation: We translate img1 to right by 1 unit and down by 1 unit.

The number of positions that have a 1 in both images is 3 (shown in red).

Example 2: Input: img1 = [[1]], img2 = [[1]] Output: 1

Example 3: Input: img1 = [[0]], img2 = [[0]] Output: 0

Constraints:

n == img1.length == img1[i].length n == img2.length == img2[i].length 1 <= n <= 30 img1[i][j] is either 0 or 1. img2[i][j] is either 0 or 1.

==
Problem Number: 896 URL: https://leetcode.com/problems/new-21-game Title: 837. New 21 Game Problem Description: Alice plays the following game, loosely based on the card game "21". Alice starts with 0 points and draws numbers while she has less than k points. During each draw, she gains an integer number of points randomly from the range [1, maxPts], where maxPts is an integer. Each draw is independent and the outcomes have equal probabilities. Alice stops drawing numbers when she gets k or more points. Return the probability that Alice has n or fewer points. Answers within 10-5 of the actual answer are considered accepted. Example 1: Input: n = 10, k = 1, maxPts = 10 Output: 1.00000 Explanation: Alice gets a single card, then stops.

Example 2: Input: n = 6, k = 1, maxPts = 10 Output: 0.60000 Explanation: Alice gets a single card, then stops. In 6 out of 10 possibilities, she is at or below 6 points.

Example 3: Input: n = 21, k = 17, maxPts = 10 Output: 0.73278

Constraints:

0 <= k <= n <= 104 1 <= maxPts <= 104

==
Problem Number: 897 URL: https://leetcode.com/problems/push-dominoes Title: 838. Push Dominoes Problem Description: There are n dominoes in a line, and we place each domino vertically upright. In the beginning, we simultaneously push some of the dominoes either to the left or to the right. After each second, each domino that is falling to the left pushes the adjacent domino on the left. Similarly, the dominoes falling to the right push their adjacent dominoes standing on the right. When a vertical domino has dominoes falling on it from both sides, it stays still due to the balance of the forces. For the purposes of this question, we will consider that a falling domino expends no additional force to a falling or already fallen domino. You are given a string dominoes representing the initial state where:

dominoes[i] = 'L', if the ith domino has been pushed to the left, dominoes[i] = 'R', if the ith domino has been pushed to the right, and dominoes[i] = '.', if the ith domino has not been pushed.

Return a string representing the final state. Example 1: Input: dominoes = ”RR.L” Output: ”RR.L” Explanation: The first domino expends no additional force on the second domino.

Example 2:

Input: dominoes = ”.L.R...LR..L..” Output: ”LL.RR.LLRRLL..”

Constraints:

n == dominoes.length 1 <= n <= 105 dominoes[i] is either ’L’, ’R’, or ’.’.

==

Problem Number: 898 URL: https://leetcode.com/problems/magic-squares-in-grid Title: 840. Magic Squares In Grid Problem Description: A 3 x 3 magic square is a 3 x 3 grid filled with distinct numbers from 1 to 9 such that each row, column, and both diagonals all have the same sum. Given a row x col grid of integers, how many 3 x 3 ”magic square” subgrids are there? (Each subgrid is contiguous). Example 1:

Input: grid = [[4,3,8,4],[9,5,1,9],[2,7,6,2]] Output: 1 Explanation: The following subgrid is a 3 x 3 magic square:

while this one is not:

In total, there is only one magic square inside the given grid.

Example 2: Input: grid = [[8]] Output: 0

Constraints:

row == grid.length col == grid[i].length 1 <= row, col <= 10 0 <= grid[i][j] <= 15

==

Problem Number: 899 URL: https://leetcode.com/problems/keys-and-rooms Title: 841. Keys and Rooms Problem Description: There are n rooms labeled from 0 to n - 1 and all the rooms are locked except for room 0. Your goal is to visit all the rooms. However, you cannot enter a locked room without having its key. When you visit a room, you may find a set of distinct keys in it. Each key has a number on it, denoting which room it unlocks, and you can take all of them with you to unlock the other rooms. Given an array rooms where rooms[i] is the set of keys that you can obtain if you visited room i, return true if you can visit all the rooms, or false otherwise. Example 1: Input: rooms = [[1],[2],[3],[]] Output: true Explanation: We visit room 0 and pick up key 1. We then visit room 1 and pick up key 2. We then visit room 2 and pick up key 3. We then visit room 3. Since we were able to visit every room, we return true.

Example 2: Input: rooms = [[1,3],[3,0,1],[2],[0]] Output: false Explanation: We can not enter room number 2 since the only key that unlocks it is in that room.

Constraints:

n == rooms.length 2 <= n <= 1000 0 <= rooms[i].length <= 1000 1 <= sum(rooms[i].length) <= 3000 0 <= rooms[i][j] < n All the values of rooms[i] are unique.

==
Problem Number: 900 URL: https://leetcode.com/problems/split-array-into-fibonacci-sequence Title: 842. Split Array into Fibonacci Sequence Problem Description: You are given a string of digits num, such as ”123456579”. We can split it into a Fibonacci-like sequence [123, 456, 579]. Formally, a Fibonacci-like sequence is a list f of non-negative integers such that:

0 <= f[i] < 231, (that is, each integer fits in a 32-bit signed integer type), f.length >= 3, and f[i] + f[i + 1] == f[i + 2] for all 0 <= i < f.length - 2.

Note that when splitting the string into pieces, each piece must not have extra leading zeroes, except if the piece is the number 0 itself. Return any Fibonacci-like sequence split from num, or return [] if it cannot be done. Example 1: Input: num = ”1101111” Output: [11,0,11,11] Explanation: The output [110, 1, 111] would also be accepted.

Example 2: Input: num = ”112358130” Output: [] Explanation: The task is impossible.

Example 3: Input: num = ”0123” Output: [] Explanation: Leading zeroes are not allowed, so ”01”, ”2”, ”3” is not valid.

Constraints:

1 <= num.length <= 200 num contains only digits.

==
Problem Number: 901 URL: https://leetcode.com/problems/longest-mountain-in-array Title: 845. Longest Mountain in Array Problem Description: You may recall that an array arr is a mountain array if and only if:

arr.length >= 3 There exists some index i (0-indexed) with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i] arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Given an integer array arr, return the length of the longest subarray, which is a mountain. Return 0 if there is no mountain subarray. Example 1: Input: arr = [2,1,4,7,3,2,5] Output: 5 Explanation: The largest mountain is [1,4,7,3,2] which has length 5.

Example 2: Input: arr = [2,2,2] Output: 0 Explanation: There is no mountain.

Constraints:

1 <= arr.length <= 104 0 <= arr[i] <= 104

Follow up:

Can you solve it using only one pass? Can you solve it in O(1) space?

==
Problem Number: 902 URL: https://leetcode.com/problems/hand-of-straights Title: 846. Hand of Straights Problem Description: Alice has some number of cards and she wants to rearrange the cards into groups so that each group is of size groupSize, and consists of groupSize consecutive cards. Given an integer array hand where hand[i] is the value written on the ith card and an integer groupSize, return true if she can rearrange the cards, or false otherwise. Example 1: Input: hand = [1,2,3,6,2,3,4,7,8], groupSize = 3 Output: true Explanation: Alice's hand can be rearranged as [1,2,3],[2,3,4],[6,7,8]

Example 2: Input: hand = [1,2,3,4,5], groupSize = 4 Output: false Explanation: Alice's hand can not be rearranged into groups of 4.

Constraints:

1 <= hand.length <= 104 0 <= hand[i] <= 109 1 <= groupSize <= hand.length

Note: This question is the same as 1296: https://leetcode.com/problems/divide-array-in-sets-of-k-consecutive-numbers/

==
Problem Number: 903 URL: https://leetcode.com/problems/shifting-letters Title: 848. Shifting Letters Problem Description: You are given a string s of lowercase English letters and an integer array shifts of the same length. Call the shift() of a letter, the next letter in the alphabet, (wrapping around so that 'z' becomes 'a').

For example, shift('a') = 'b', shift('t') = 'u', and shift('z') = 'a'.

Now for each shifts[i] = x, we want to shift the first i + 1 letters of s, x times. Return the final string after all such shifts to s are applied. Example 1: Input: s = "abc", shifts = [3,5,9] Output: "rpl" Explanation: We start with "abc". After shifting the first 1 letters of s by 3, we have "dbc". After shifting the first 2 letters of s by 5, we have "igc". After shifting the first 3 letters of s by 9, we have "rpl", the answer.

Example 2: Input: s = "aaa", shifts = [1,2,3] Output: "gfd"

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters. shifts.length == s.length 0 <= shifts[i] <= 109

==
Problem Number: 904 URL: https://leetcode.com/problems/maximize-distance-to-closest-person Title: 849. Maximize Distance to Closest Person Problem Description: You are given an array representing a row of seats where seats[i] = 1 represents a person sitting in the ith seat, and seats[i] = 0

represents that the ith seat is empty (0-indexed). There is at least one empty seat, and at least one person sitting. Alex wants to sit in the seat such that the distance between him and the closest person to him is maximized. Return that maximum distance to the closest person. Example 1:

Input: seats = [1,0,0,0,1,0,1] Output: 2 Explanation: If Alex sits in the second open seat (i.e. seats[2]), then the closest person has distance 2. If Alex sits in any other open seat, the closest person has distance 1. Thus, the maximum distance to the closest person is 2.

Example 2: Input: seats = [1,0,0,0] Output: 3 Explanation: If Alex sits in the last seat (i.e. seats[3]), the closest person is 3 seats away. This is the maximum distance possible, so the answer is 3.

Example 3: Input: seats = [0,1] Output: 1

Constraints:

2 <= seats.length <= 2 * 104 seats[i] is 0 or 1. At least one seat is empty. At least one seat is occupied.

==
Problem Number: 905 URL: https://leetcode.com/problems/loud-and-rich Title: 851. Loud and Rich Problem Description: There is a group of n people labeled from 0 to n - 1 where each person has a different amount of money and a different level of quietness. You are given an array richer where richer[i] = [ai, bi] indicates that ai has more money than bi and an integer array quiet where quiet[i] is the quietness of the ith person. All the given data in richer are logically correct (i.e., the data will not lead you to a situation where x is richer than y and y is richer than x at the same time). Return an integer array answer where answer[x] = y if y is the least quiet person (that is, the person y with the smallest value of quiet[y]) among all people who definitely have equal to or more money than the person x. Example 1: Input: richer = [[1,0],[2,1],[3,1],[3,7],[4,3],[5,3],[6,3]], quiet = [3,2,5,4,6,1,7,0] Output: [5,5,2,5,4,5,6,7] Explanation: answer[0] = 5. Person 5 has more money than 3, which has more money than 1, which has more money than 0. The only person who is quieter (has lower quiet[x]) is person 7, but it is not clear if they have more money than person 0. answer[7] = 7. Among all people that definitely have equal to or more money than person 7 (which could be persons 3, 4, 5, 6, or 7), the person who is the quietest (has lower quiet[x]) is person 7. The other answers can be filled out with similar reasoning.

Example 2: Input: richer = [], quiet = [0] Output: [0]

Constraints:

n == quiet.length 1 <= n <= 500 0 <= quiet[i] < n All the values of quiet are unique. 0 <= richer.length <= n * (n - 1) / 2 0 <= ai, bi < n ai != bi All the pairs of richer are unique. The observations in richer are all logically consistent.

== Problem Number: 906 URL: https://leetcode.com/problems/peak-index-in-a-mountain-array Title: 852. Peak Index in a Mountain Array Problem Description: An array arr is a mountain if the following properties hold:

arr.length >= 3 There exists some i with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i] arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Given a mountain array arr, return the index i such that arr[0] < arr[1] < ... < arr[i - 1] < arr[i] > arr[i + 1] > ... > arr[arr.length - 1]. You must solve it in O(log(arr.length)) time complexity. Example 1: Input: arr = [0,1,0] Output: 1

Example 2: Input: arr = [0,2,1,0] Output: 1

Example 3: Input: arr = [0,10,5,2] Output: 1

Constraints:

3 <= arr.length <= 105 0 <= arr[i] <= 106 arr is guaranteed to be a mountain array.

== Problem Number: 907 URL: https://leetcode.com/problems/car-fleet Title: 853. Car Fleet Problem Description: There are n cars going to the same destination along a one-lane road. The destination is target miles away. You are given two integer array position and speed, both of length n, where position[i] is the position of the ith car and speed[i] is the speed of the ith car (in miles per hour). A car can never pass another car ahead of it, but it can catch up to it and drive bumper to bumper at the same speed. The faster car will slow down to match the slower car's speed. The distance between these two cars is ignored (i.e., they are assumed to have the same position). A car fleet is some non-empty set of cars driving at the same position and same speed. Note that a single car is also a car fleet. If a car catches up to a car fleet right at the destination point, it will still be considered as one car fleet. Return the number of car fleets that will arrive at the destination. Example 1: Input: target = 12, position = [10,8,0,5,3], speed = [2,4,1,1,3] Output: 3 Explanation: The cars starting at 10 (speed 2) and 8 (speed 4) become a fleet, meeting each other at 12. The car starting at 0 does not catch up to any other car, so it is a fleet by itself. The cars starting at 5 (speed 1) and 3 (speed 3) become a fleet, meeting each other at 6. The fleet moves at speed 1 until it reaches target. Note that no other cars meet these fleets before the destination, so the answer is 3.

Example 2: Input: target = 10, position = [3], speed = [3] Output: 1 Explanation: There is only one car, hence there is only one fleet.

Example 3: Input: target = 100, position = [0,2,4], speed = [4,2,1] Output: 1 Explanation: The cars starting at 0 (speed 4) and 2 (speed 2) become a fleet, meeting each other at 4. The fleet moves at speed 2. Then, the fleet (speed 2) and the car starting at 4 (speed 1) become one fleet, meeting each other at 6. The fleet moves at speed 1 until it reaches target.

Constraints:

n == position.length == speed.length 1 <= n <= 105 0 < target <= 106 0 <= position[i] < target All the values of position are unique. 0 < speed[i] <= 106

==
Problem Number: 908 URL: https://leetcode.com/problems/exam-room Title: 855. Exam Room Problem Description: There is an exam room with n seats in a single row labeled from 0 to n - 1. When a student enters the room, they must sit in the seat that maximizes the distance to the closest person. If there are multiple such seats, they sit in the seat with the lowest number. If no one is in the room, then the student sits at seat number 0. Design a class that simulates the mentioned exam room. Implement the ExamRoom class:

ExamRoom(int n) Initializes the object of the exam room with the number of the seats n. int seat() Returns the label of the seat at which the next student will set. void leave(int p) Indicates that the student sitting at seat p will leave the room. It is guaranteed that there will be a student sitting at seat p.

Example 1: Input ["ExamRoom", "seat", "seat", "seat", "seat", "leave", "seat"] [[10], [], [], [], [], [4], []] Output [null, 0, 9, 4, 2, null, 5]

Explanation ExamRoom examRoom = new ExamRoom(10); examRoom.seat(); // return 0, no one is in the room, then the student sits at seat number 0. examRoom.seat(); // return 9, the student sits at the last seat number 9. examRoom.seat(); // return 4, the student sits at the last seat number 4. examRoom.seat(); // return 2, the student sits at the last seat number 2. examRoom.leave(4); examRoom.seat(); // return 5, the student sits at the last seat number 5.

Constraints:

1 <= n <= 109 It is guaranteed that there is a student sitting at seat p. At most 104 calls will be made to seat and leave.

==
Problem Number: 909 URL: https://leetcode.com/problems/score-of-parentheses Title: 856. Score of Parentheses Problem Description: Given a balanced parentheses string s, return the score of the string. The score of a balanced parentheses string is based on the following rule:

"()" has score 1. AB has score A + B, where A and B are balanced parentheses strings. (A) has score 2 * A, where A is a balanced parentheses string.

Example 1: Input: s = "()" Output: 1

Example 2: Input: s = "(())" Output: 2

Example 3: Input: s = ”()()” Output: 2

Constraints:

2 <= s.length <= 50 s consists of only ’(’ and ’)’. s is a balanced parentheses string.

==
Problem Number: 910 URL: https://leetcode.com/problems/mirror-reflection Title: 858. Mirror Reflection Problem Description: There is a special square room with mirrors on each of the four walls. Except for the southwest corner, there are receptors on each of the remaining corners, numbered 0, 1, and 2. The square room has walls of length p and a laser ray from the southwest corner first meets the east wall at a distance q from the 0th receptor. Given the two integers p and q, return the number of the receptor that the ray meets first. The test cases are guaranteed so that the ray will meet a receptor eventually. Example 1:

Input: p = 2, q = 1 Output: 2 Explanation: The ray meets receptor 2 the first time it gets reflected back to the left wall.

Example 2: Input: p = 3, q = 1 Output: 1

Constraints:

1 <= q <= p <= 1000

==
Problem Number: 911 URL: https://leetcode.com/problems/score-after-flipping-matrix Title: 861. Score After Flipping Matrix Problem Description: You are given an m x n binary matrix grid. A move consists of choosing any row or column and toggling each value in that row or column (i.e., changing all 0’s to 1’s, and all 1’s to 0’s). Every row of the matrix is interpreted as a binary number, and the score of the matrix is the sum of these numbers. Return the highest possible score after making any number of moves (including zero moves). Example 1:

Input: grid = [[0,0,1,1],[1,0,1,0],[1,1,0,0]] Output: 39 Explanation: 0b1111 + 0b1001 + 0b1111 = 15 + 9 + 15 = 39

Example 2: Input: grid = [[0]] Output: 1

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 20 grid[i][j] is either 0 or 1.

==
Problem Number: 912 URL: https://leetcode.com/problems/all-nodes-distance-k-in-binary-tree Title: 863. All Nodes Distance K in Binary Tree Problem Description: Given the root of a binary tree, the value of a target node target,

and an integer k, return an array of the values of all nodes that have a distance k from the target node. You can return the answer in any order.   Example 1:

Input: root = [3,5,1,6,2,0,8,null,null,7,4], target = 5, k = 2 Output: [7,4,1] Explanation: The nodes that are a distance 2 from the target node (with value 5) have values 7, 4, and 1.

Example 2: Input: root = [1], target = 1, k = 3 Output: []

Constraints:

The number of nodes in the tree is in the range [1, 500]. 0 <= Node.val <= 500 All the values Node.val are unique. target is the value of one of the nodes in the tree. 0 <= k <= 1000

==

Problem Number: 913 URL: https://leetcode.com/problems/smallest-subtree-with-all-the-deepest-nodes Title: 865. Smallest Subtree with all the Deepest Nodes Problem Description: Given the root of a binary tree, the depth of each node is the shortest distance to the root. Return the smallest subtree such that it contains all the deepest nodes in the original tree. A node is called the deepest if it has the largest depth possible among any node in the entire tree. The subtree of a node is a tree consisting of that node, plus the set of all descendants of that node.   Example 1:

Input: root = [3,5,1,6,2,0,8,null,null,7,4] Output: [2,7,4] Explanation: We return the node with value 2, colored in yellow in the diagram. The nodes coloured in blue are the deepest nodes of the tree. Notice that nodes 5, 3 and 2 contain the deepest nodes in the tree but node 2 is the smallest subtree among them, so we return it.

Example 2: Input: root = [1] Output: [1] Explanation: The root is the deepest node in the tree.

Example 3: Input: root = [0,1,3,null,2] Output: [2] Explanation: The deepest node in the tree is 2, the valid subtrees are the subtrees of nodes 2, 1 and 0 but the subtree of node 2 is the smallest.

Constraints:

The number of nodes in the tree will be in the range [1, 500]. 0 <= Node.val <= 500 The values of the nodes in the tree are unique.

Note: This question is the same as 1123: https://leetcode.com/problems/lowest-common-ancestor-of-deepest-leaves/

==

Problem Number: 914 URL: https://leetcode.com/problems/prime-palindrome Title: 866. Prime Palindrome Problem Description: Given an integer n, return the smallest prime palindrome greater than or equal to n. An integer is prime if it has exactly two divisors: 1 and itself. Note that 1 is not a prime number.

For example, 2, 3, 5, 7, 11, and 13 are all primes.

An integer is a palindrome if it reads the same from left to right as it does from right to left.

For example, 101 and 12321 are palindromes.

The test cases are generated so that the answer always exists and is in the range [2, 2 * 108]. Example 1: Input: n = 6 Output: 7 Example 2: Input: n = 8 Output: 11 Example 3: Input: n = 13 Output: 101

Constraints:

1 <= n <= 108
==
Problem Number: 915 URL: https://leetcode.com/problems/reordered-power-of-2 Title: 869. Reordered Power of 2 Problem Description: You are given an integer n. We reorder the digits in any order (including the original order) such that the leading digit is not zero. Return true if and only if we can do this so that the resulting number is a power of two. Example 1: Input: n = 1 Output: true

Example 2: Input: n = 10 Output: false

Constraints:

1 <= n <= 109
==
Problem Number: 916 URL: https://leetcode.com/problems/advantage-shuffle Title: 870. Advantage Shuffle Problem Description: You are given two integer arrays nums1 and nums2 both of the same length. The advantage of nums1 with respect to nums2 is the number of indices i for which nums1[i] > nums2[i]. Return any permutation of nums1 that maximizes its advantage with respect to nums2. Example 1: Input: nums1 = [2,7,11,15], nums2 = [1,10,4,11] Output: [2,11,7,15] Example 2: Input: nums1 = [12,24,8,32], nums2 = [13,25,32,11] Output: [24,32,8,12]

Constraints:

1 <= nums1.length <= 105 nums2.length == nums1.length 0 <= nums1[i], nums2[i] <= 109
==
Problem Number: 917 URL: https://leetcode.com/problems/length-of-longest-fibonacci-subsequence Title: 873. Length of Longest Fibonacci Subsequence Problem Description: A sequence x1, x2, ..., xn is Fibonacci-like if:

n >= 3 xi + xi+1 == xi+2 for all i + 2 <= n

Given a strictly increasing array arr of positive integers forming a sequence, return the length of the longest Fibonacci-like subsequence of arr. If one does not exist, return 0. A subsequence is derived from another sequence arr by deleting any number of elements (including none) from arr, without changing the order of the remaining elements. For example, [3, 5, 8] is a subsequence of [3, 4, 5, 6, 7, 8]. Example 1: Input: arr = [1,2,3,4,5,6,7,8] Output: 5 Explanation: The longest subsequence that is fibonacci-like: [1,2,3,5,8]. Example 2: Input: arr = [1,3,7,11,12,14,18] Output: 3 Explanation: The longest subsequence that is fibonacci-like: [1,11,12], [3,11,14] or [7,11,18]. Constraints:

3 <= arr.length <= 1000 1 <= arr[i] < arr[i + 1] <= 109

==

Problem Number: 918 URL: https://leetcode.com/problems/walking-robot-simulation Title: 874. Walking Robot Simulation Problem Description: A robot on an infinite XY-plane starts at point (0, 0) facing north. The robot can receive a sequence of these three possible types of commands:

-2: Turn left 90 degrees. -1: Turn right 90 degrees. 1 <= k <= 9: Move forward k units, one unit at a time.

Some of the grid squares are obstacles. The ith obstacle is at grid point obstacles[i] = (xi, yi). If the robot runs into an obstacle, then it will instead stay in its current location and move on to the next command. Return the maximum Euclidean distance that the robot ever gets from the origin squared (i.e. if the distance is 5, return 25). Note:

North means +Y direction. East means +X direction. South means -Y direction. West means -X direction. There can be obstacle in [0,0].

Example 1: Input: commands = [4,-1,3], obstacles = [] Output: 25 Explanation: The robot starts at (0, 0): 1. Move north 4 units to (0, 4). 2. Turn right. 3. Move east 3 units to (3, 4). The furthest point the robot ever gets from the origin is (3, 4), which squared is 32 + 42 = 25 units away.

Example 2: Input: commands = [4,-1,4,-2,4], obstacles = [[2,4]] Output: 65 Explanation: The robot starts at (0, 0): 1. Move north 4 units to (0, 4). 2. Turn right. 3. Move east 1 unit and get blocked by the obstacle at (2, 4), robot is at (1, 4). 4. Turn left. 5. Move north 4 units to (1, 8). The furthest point the robot ever gets from the origin is (1, 8), which squared is 12 + 82 = 65 units away.

Example 3: Input: commands = [6,-1,-1,6], obstacles = [] Output: 36 Explanation: The robot starts at (0, 0): 1. Move north 6 units to (0, 6). 2. Turn right. 3. Turn right. 4. Move south 6 units to (0, 0). The furthest point the robot ever gets from the origin is (0, 6), which squared is 62 = 36 units away.

Constraints:

1 <= commands.length <= 104 commands[i] is either -2, -1, or an integer in the range [1, 9]. 0 <= obstacles.length <= 104 -3 * 104 <= xi, yi <= 3 * 104 The answer is guaranteed to be less than 231.

==
Problem Number: 919 URL: https://leetcode.com/problems/koko-eating-bananas Title: 875. Koko Eating Bananas Problem Description: Koko loves to eat bananas. There are n piles of bananas, the ith pile has piles[i] bananas. The guards have gone and will come back in h hours. Koko can decide her bananas-per-hour eating speed of k. Each hour, she chooses some pile of bananas and eats k bananas from that pile. If the pile has less than k bananas, she eats all of them instead and will not eat any more bananas during this hour. Koko likes to eat slowly but still wants to finish eating all the bananas before the guards return. Return the minimum integer k such that she can eat all the bananas within h hours. Example 1: Input: piles = [3,6,7,11], h = 8 Output: 4

Example 2: Input: piles = [30,11,23,4,20], h = 5 Output: 30

Example 3: Input: piles = [30,11,23,4,20], h = 6 Output: 23

Constraints:

1 <= piles.length <= 104 piles.length <= h <= 109 1 <= piles[i] <= 109

==
Problem Number: 920 URL: https://leetcode.com/problems/stone-game Title: 877. Stone Game Problem Description: Alice and Bob play a game with piles of stones. There are an even number of piles arranged in a row, and each pile has a positive integer number of stones piles[i]. The objective of the game is to end with the most stones. The total number of stones across all the piles is odd, so there are no ties. Alice and Bob take turns, with Alice starting first. Each turn, a player takes the entire pile of stones either from the beginning or from the end of the row. This continues until there are no more piles left, at which point the person with the most stones wins. Assuming Alice and Bob play optimally, return true if Alice wins the game, or false if Bob wins. Example 1: Input: piles = [5,3,4,5] Output: true Explanation: Alice starts first, and can only take the first 5 or the last 5. Say she takes the first 5, so that the row becomes [3, 4, 5]. If Bob takes 3, then the board is [4, 5], and Alice takes 5 to win with 10 points. If Bob takes the last 5, then the board is [3, 4], and Alice takes 4 to win with 9 points. This demonstrated that taking the first 5 was a winning move for Alice, so we return true.

Example 2: Input: piles = [3,7,2,3] Output: true

Constraints:

2 <= piles.length <= 500 piles.length is even. 1 <= piles[i] <= 500 sum(piles[i]) is odd.

==
Problem Number: 921 URL: https://leetcode.com/problems/decoded-string-at-index Title: 880. Decoded String at Index Problem Description: You are

given an encoded string s. To decode the string to a tape, the encoded string is read one character at a time and the following steps are taken:

If the character read is a letter, that letter is written onto the tape. If the character read is a digit d, the entire current tape is repeatedly written d - 1 more times in total.

Given an integer k, return the kth letter (1-indexed) in the decoded string. Example 1: Input: s = "leet2code3", k = 10 Output: "o" Explanation: The decoded string is "leetleetcodeleetleetcodeleetleetcode". The 10th letter in the string is "o".

Example 2: Input: s = "ha22", k = 5 Output: "h" Explanation: The decoded string is "hahahaha". The 5th letter is "h".

Example 3: Input: s = "a2345678999999999999999", k = 1 Output: "a" Explanation: The decoded string is "a" repeated 8301530446056247680 times. The 1st letter is "a".

Constraints:

2 <= s.length <= 100 s consists of lowercase English letters and digits 2 through 9. s starts with a letter. 1 <= k <= 109 It is guaranteed that k is less than or equal to the length of the decoded string. The decoded string is guaranteed to have less than 263 letters.

==
Problem Number: 922 URL: https://leetcode.com/problems/boats-to-save-people Title: 881. Boats to Save People Problem Description: You are given an array people where people[i] is the weight of the ith person, and an infinite number of boats where each boat can carry a maximum weight of limit. Each boat carries at most two people at the same time, provided the sum of the weight of those people is at most limit. Return the minimum number of boats to carry every given person. Example 1: Input: people = [1,2], limit = 3 Output: 1 Explanation: 1 boat (1, 2)

Example 2: Input: people = [3,2,2,1], limit = 3 Output: 3 Explanation: 3 boats (1, 2), (2) and (3)

Example 3: Input: people = [3,5,3,4], limit = 5 Output: 4 Explanation: 4 boats (3), (3), (4), (5)

Constraints:

1 <= people.length <= 5 * 104 1 <= people[i] <= limit <= 3 * 104

==
Problem Number: 923 URL: https://leetcode.com/problems/spiral-matrix-iii Title: 885. Spiral Matrix III Problem Description: You start at the cell (rStart, cStart) of an rows x cols grid facing east. The northwest corner is at the first row and column in the grid, and the southeast corner is at the last row and column. You will walk in a clockwise spiral shape to visit every position in

this grid. Whenever you move outside the grid's boundary, we continue our walk outside the grid (but may return to the grid boundary later.). Eventually, we reach all rows * cols spaces of the grid. Return an array of coordinates representing the positions of the grid in the order you visited them. Example 1:

Input: rows = 1, cols = 4, rStart = 0, cStart = 0 Output: [[0,0],[0,1],[0,2],[0,3]]

Example 2:

Input: rows = 5, cols = 6, rStart = 1, cStart = 4 Output: [[1,4],[1,5],[2,5],[2,4],[2,3],[1,3],[0,3],[0,4],[0,5],[3,5],[3,4],

Constraints:

1 <= rows, cols <= 100 0 <= rStart < rows 0 <= cStart < cols

==
Problem Number: 924 URL: https://leetcode.com/problems/possible-bipartition Title: 886. Possible Bipartition Problem Description: We want to split a group of n people (labeled from 1 to n) into two groups of any size. Each person may dislike some other people, and they should not go into the same group. Given the integer n and the array dislikes where dislikes[i] = [ai, bi] indicates that the person labeled ai does not like the person labeled bi, return true if it is possible to split everyone into two groups in this way. Example 1: Input: n = 4, dislikes = [[1,2],[1,3],[2,4]] Output: true Explanation: The first group has [1,4], and the second group has [2,3].

Example 2: Input: n = 3, dislikes = [[1,2],[1,3],[2,3]] Output: false Explanation: We need at least 3 groups to divide them. We cannot put them in two groups.

Constraints:

1 <= n <= 2000 0 <= dislikes.length <= 104 dislikes[i].length == 2 1 <= ai < bi <= n All the pairs of dislikes are unique.

==
Problem Number: 925 URL: https://leetcode.com/problems/construct-binary-tree-from-preorder-and-postorder-traversal Title: 889. Construct Binary Tree from Preorder and Postorder Traversal Problem Description: Given two integer arrays, preorder and postorder where preorder is the preorder traversal of a binary tree of distinct values and postorder is the postorder traversal of the same tree, reconstruct and return the binary tree. If there exist multiple answers, you can return any of them. Example 1:

Input: preorder = [1,2,4,5,3,6,7], postorder = [4,5,2,6,7,3,1] Output: [1,2,3,4,5,6,7]

Example 2: Input: preorder = [1], postorder = [1] Output: [1]

Constraints:

1 <= preorder.length <= 30 1 <= preorder[i] <= preorder.length All the values of preorder are unique. postorder.length == preorder.length 1 <= postorder[i]

<= postorder.length All the values of postorder are unique. It is guaranteed that preorder and postorder are the preorder traversal and postorder traversal of the same binary tree.

==
Problem Number: 926 URL: https://leetcode.com/problems/find-and-replace-pattern Title: 890. Find and Replace Pattern Problem Description: Given a list of strings words and a string pattern, return a list of words[i] that match pattern. You may return the answer in any order. A word matches the pattern if there exists a permutation of letters p so that after replacing every letter x in the pattern with p(x), we get the desired word. Recall that a permutation of letters is a bijection from letters to letters: every letter maps to another letter, and no two letters map to the same letter. Example 1: Input: words = ["abc","deq","mee","aqq","dkd","ccc"], pattern = "abb" Output: ["mee","aqq"] Explanation: "mee" matches the pattern because there is a permutation {a -> m, b -> e, ...}. "ccc" does not match the pattern because {a -> c, b -> c, ...} is not a permutation, since a and b map to the same letter.

Example 2: Input: words = ["a","b","c"], pattern = "a" Output: ["a","b","c"]

Constraints:

1 <= pattern.length <= 20 1 <= words.length <= 50 words[i].length == pattern.length pattern and words[i] are lowercase English letters.

==
Problem Number: 927 URL: https://leetcode.com/problems/groups-of-special-equivalent-strings Title: 893. Groups of Special-Equivalent Strings Problem Description: You are given an array of strings of the same length words. In one move, you can swap any two even indexed characters or any two odd indexed characters of a string words[i]. Two strings words[i] and words[j] are special-equivalent if after any number of moves, words[i] == words[j].

For example, words[i] = "zzxy" and words[j] = "xyzz" are special-equivalent because we may make the moves "zzxy" -> "xzzy" -> "xyzz".

A group of special-equivalent strings from words is a non-empty subset of words such that:

Every pair of strings in the group are special equivalent, and The group is the largest size possible (i.e., there is not a string words[i] not in the group such that words[i] is special-equivalent to every string in the group).

Return the number of groups of special-equivalent strings from words. Example 1: Input: words = ["abcd","cdab","cbad","xyzz","zzxy","zzyx"] Output: 3 Explanation: One group is ["abcd", "cdab", "cbad"], since they are all pairwise special equivalent, and none of the other strings is all pairwise special equivalent to these. The other two groups are ["xyzz", "zzxy"] and ["zzyx"]. Note that in particular, "zzxy" is not special equivalent to "zzyx".

Example 2: Input: words = ["abc","acb","bac","bca","cab","cba"] Output: 3

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 20 words[i] consist of lowercase English letters. All the strings are of the same length.

==
Problem Number: 928 URL: https://leetcode.com/problems/all-possible-full-binary-trees Title: 894. All Possible Full Binary Trees Problem Description: Given an integer n, return a list of all possible full binary trees with n nodes. Each node of each tree in the answer must have Node.val == 0. Each element of the answer is the root node of one possible tree. You may return the final list of trees in any order. A full binary tree is a binary tree where each node has exactly 0 or 2 children. Example 1:

Input: n = 7 Output: [[0,0,0,null,null,0,0,null,null,0,0],[0,0,0,null,null,0,0,0,0],[0,0,0,0,0,0,0],[0,0,0,0,0,null,null,nu

Example 2: Input: n = 3 Output: [[0,0,0]]

Constraints:

1 <= n <= 20

==
Problem Number: 929 URL: https://leetcode.com/problems/bitwise-ors-of-subarrays Title: 898. Bitwise ORs of Subarrays Problem Description: Given an integer array arr, return the number of distinct bitwise ORs of all the non-empty subarrays of arr. The bitwise OR of a subarray is the bitwise OR of each integer in the subarray. The bitwise OR of a subarray of one integer is that integer. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: arr = [0] Output: 1 Explanation: There is only one possible result: 0.

Example 2: Input: arr = [1,1,2] Output: 3 Explanation: The possible subarrays are [1], [1], [2], [1, 1], [1, 2], [1, 1, 2]. These yield the results 1, 1, 2, 1, 3, 3. There are 3 unique values, so the answer is 3.

Example 3: Input: arr = [1,2,4] Output: 6 Explanation: The possible results are 1, 2, 3, 4, 6, and 7.

Constraints:

1 <= arr.length <= 5 * 104 0 <= arr[i] <= 109

==
Problem Number: 930 URL: https://leetcode.com/problems/rle-iterator Title: 900. RLE Iterator Problem Description: We can use run-length encoding (i.e., RLE) to encode a sequence of integers. In a run-length encoded array of even length encoding (0-indexed), for all even i, encoding[i] tells us the number of times that the non-negative integer value encoding[i + 1] is repeated in the sequence.

For example, the sequence arr = [8,8,8,5,5] can be encoded to be encoding = [3,8,2,5]. encoding = [3,8,0,9,2,5] and encoding = [2,8,1,8,2,5] are also valid RLE of arr.

Given a run-length encoded array, design an iterator that iterates through it. Implement the RLEIterator class:

RLEIterator(int[] encoded) Initializes the object with the encoded array encoded. int next(int n) Exhausts the next n elements and returns the last element exhausted in this way. If there is no element left to exhaust, return -1 instead.

Example 1: Input ["RLEIterator", "next", "next", "next", "next"] [[[3, 8, 0, 9, 2, 5]], [2], [1], [1], [2]] Output [null, 8, 8, 5, -1]

Explanation RLEIterator rLEIterator = new RLEIterator([3, 8, 0, 9, 2, 5]); // This maps to the sequence [8,8,8,5,5]. rLEIterator.next(2); // exhausts 2 terms of the sequence, returning 8. The remaining sequence is now [8, 5, 5]. rLEIterator.next(1); // exhausts 1 term of the sequence, returning 8. The remaining sequence is now [5, 5]. rLEIterator.next(1); // exhausts 1 term of the sequence, returning 5. The remaining sequence is now [5]. rLEIterator.next(2); // exhausts 2 terms, returning -1. This is because the first term exhausted was 5, but the second term did not exist. Since the last term exhausted does not exist, we return -1.

Constraints:

2 <= encoding.length <= 1000 encoding.length is even. 0 <= encoding[i] <= 109 1 <= n <= 109 At most 1000 calls will be made to next.

==
Problem Number: 931 URL: https://leetcode.com/problems/online-stock-span Title: 901. Online Stock Span Problem Description: Design an algorithm that collects daily price quotes for some stock and returns the span of that stock's price for the current day. The span of the stock's price in one day is the maximum number of consecutive days (starting from that day and going backward) for which the stock price was less than or equal to the price of that day.

For example, if the prices of the stock in the last four days is [7,2,1,2] and the price of the stock today is 2, then the span of today is 4 because starting from today, the price of the stock was less than or equal 2 for 4 consecutive days. Also, if the prices of the stock in the last four days is [7,34,1,2] and the price of the stock today is 8, then the span of today is 3 because starting from today, the price of the stock was less than or equal 8 for 3 consecutive days.

Implement the StockSpanner class:

StockSpanner() Initializes the object of the class. int next(int price) Returns the span of the stock's price given that today's price is price.

Example 1: Input ["StockSpanner", "next", "next", "next", "next", "next",

"next", "next"] [[], [100], [80], [60], [70], [60], [75], [85]] Output [null, 1, 1, 1, 2, 1, 4, 6]

Explanation StockSpanner stockSpanner = new StockSpanner(); stockSpanner.next(100); // return 1 stockSpanner.next(80); // return 1 stockSpanner.next(60); // return 1 stockSpanner.next(70); // return 2 stockSpanner.next(60); // return 1 stockSpanner.next(75); // return 4, because the last 4 prices (including today's price of 75) were less than or equal to today's price. stockSpanner.next(85); // return 6

Constraints:

1 <= price <= 105 At most 104 calls will be made to next.

==
Problem Number: 932 URL: https://leetcode.com/problems/fruit-into-baskets Title: 904. Fruit Into Baskets Problem Description: You are visiting a farm that has a single row of fruit trees arranged from left to right. The trees are represented by an integer array fruits where fruits[i] is the type of fruit the ith tree produces. You want to collect as much fruit as possible. However, the owner has some strict rules that you must follow:

You only have two baskets, and each basket can only hold a single type of fruit. There is no limit on the amount of fruit each basket can hold. Starting from any tree of your choice, you must pick exactly one fruit from every tree (including the start tree) while moving to the right. The picked fruits must fit in one of your baskets. Once you reach a tree with fruit that cannot fit in your baskets, you must stop.

Given the integer array fruits, return the maximum number of fruits you can pick. Example 1: Input: fruits = [1,2,1] Output: 3 Explanation: We can pick from all 3 trees.

Example 2: Input: fruits = [0,1,2,2] Output: 3 Explanation: We can pick from trees [1,2,2]. If we had started at the first tree, we would only pick from trees [0,1].

Example 3: Input: fruits = [1,2,3,2,2] Output: 4 Explanation: We can pick from trees [2,3,2,2]. If we had started at the first tree, we would only pick from trees [1,2].

Constraints:

1 <= fruits.length <= 105 0 <= fruits[i] < fruits.length

==
Problem Number: 933 URL: https://leetcode.com/problems/sum-of-subarray-minimums Title: 907. Sum of Subarray Minimums Problem Description: Given an array of integers arr, find the sum of min(b), where b ranges over every (contiguous) subarray of arr. Since the answer may be large, return the answer modulo 109 + 7. Example 1: Input: arr = [3,1,2,4] Output: 17

Explanation: Subarrays are [3], [1], [2], [4], [3,1], [1,2], [2,4], [3,1,2], [1,2,4], [3,1,2,4]. Minimums are 3, 1, 2, 4, 1, 1, 2, 1, 1, 1. Sum is 17.

Example 2: Input: arr = [11,81,94,43,3] Output: 444

Constraints:

1 <= arr.length <= 3 * 104 1 <= arr[i] <= 3 * 104

==

Problem Number: 934 URL: https://leetcode.com/problems/snakes-and-ladders Title: 909. Snakes and Ladders Problem Description: You are given an n x n integer matrix board where the cells are labeled from 1 to n2 in a Boustrophedon style starting from the bottom left of the board (i.e. board[n - 1][0]) and alternating direction each row. You start on square 1 of the board. In each move, starting from square curr, do the following:

Choose a destination square next with a label in the range [curr + 1, min(curr + 6, n2)].

This choice simulates the result of a standard 6-sided die roll: i.e., there are always at most 6 destinations, regardless of the size of the board.

If next has a snake or ladder, you must move to the destination of that snake or ladder. Otherwise, you move to next. The game ends when you reach the square n2.

A board square on row r and column c has a snake or ladder if board[r][c] != -1. The destination of that snake or ladder is board[r][c]. Squares 1 and n2 do not have a snake or ladder. Note that you only take a snake or ladder at most once per move. If the destination to a snake or ladder is the start of another snake or ladder, you do not follow the subsequent snake or ladder.

For example, suppose the board is [[-1,4],[-1,3]], and on the first move, your destination square is 2. You follow the ladder to square 3, but do not follow the subsequent ladder to 4.

Return the least number of moves required to reach the square n2. If it is not possible to reach the square, return -1. Example 1:

Input: board = [[-1,-1,-1,-1,-1,-1],[-1,-1,-1,-1,-1,-1],[-1,-1,-1,-1,-1,-1],[-1,35,-1,-1,13,-1],[-1,-1,-1,-1,-1,-1],[-1,15,-1,-1,-1,-1]] Output: 4 Explanation: In the beginning, you start at square 1 (at row 5, column 0). You decide to move to square 2 and must take the ladder to square 15. You then decide to move to square 17 and must take the snake to square 13. You then decide to move to square 14 and must take the ladder to square 35. You then decide to move to square 36, ending the game. This is the lowest possible number of moves to reach the last square, so return 4.

Example 2: Input: board = [[-1,-1],[-1,3]] Output: 1

Constraints:

n == board.length == board[i].length 2 <= n <= 20 board[i][j] is either -1 or in the range [1, n2]. The squares labeled 1 and n2 do not have any ladders or snakes.

===

Problem Number: 935 URL: https://leetcode.com/problems/smallest-range-ii Title: 910. Smallest Range II Problem Description: You are given an integer array nums and an integer k. For each index i where 0 <= i < nums.length, change nums[i] to be either nums[i] + k or nums[i] - k. The score of nums is the difference between the maximum and minimum elements in nums. Return the minimum score of nums after changing the values at each index.   Example 1: Input: nums = [1], k = 0 Output: 0 Explanation: The score is max(nums) - min(nums) = 1 - 1 = 0.

Example 2: Input: nums = [0,10], k = 2 Output: 6 Explanation: Change nums to be [2, 8]. The score is max(nums) - min(nums) = 8 - 2 = 6.

Example 3: Input: nums = [1,3,6], k = 3 Output: 3 Explanation: Change nums to be [4, 6, 3]. The score is max(nums) - min(nums) = 6 - 3 = 3.

Constraints:

1 <= nums.length <= 104 0 <= nums[i] <= 104 0 <= k <= 104

===

Problem Number: 936 URL: https://leetcode.com/problems/online-election Title: 911. Online Election Problem Description: You are given two integer arrays persons and times. In an election, the ith vote was cast for persons[i] at time times[i]. For each query at a time t, find the person that was leading the election at time t. Votes cast at time t will count towards our query. In the case of a tie, the most recent vote (among tied candidates) wins. Implement the TopVotedCandidate class:

TopVotedCandidate(int[] persons, int[] times) Initializes the object with the persons and times arrays. int q(int t) Returns the number of the person that was leading the election at time t according to the mentioned rules.

Example 1: Input ["TopVotedCandidate", "q", "q", "q", "q", "q", "q"] [[[0, 1, 1, 0, 0, 1, 0], [0, 5, 10, 15, 20, 25, 30]], [3], [12], [25], [15], [24], [8]] Output [null, 0, 1, 1, 0, 0, 1]

Explanation TopVotedCandidate topVotedCandidate = new TopVotedCandidate([0, 1, 1, 0, 0, 1, 0], [0, 5, 10, 15, 20, 25, 30]); topVotedCandidate.q(3); // return 0, At time 3, the votes are [0], and 0 is leading. topVotedCandidate.q(12); // return 1, At time 12, the votes are [0,1,1], and 1 is leading. topVotedCandidate.q(25); // return 1, At time 25, the votes are [0,1,1,0,0,1], and 1 is leading (as ties go to the most recent vote.) topVotedCandidate.q(15); // return 0 topVotedCandidate.q(24); // return 0 topVotedCandidate.q(8); // return 1

Constraints:

1 <= persons.length <= 5000 times.length == persons.length 0 <= persons[i] < persons.length 0 <= times[i] <= 109 times is sorted in a strictly increasing order. times[0] <= t <= 109 At most 104 calls will be made to q.

==

Problem Number: 937 URL: https://leetcode.com/problems/sort-an-array Title: 912. Sort an Array Problem Description: Given an array of integers nums, sort the array in ascending order and return it. You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible. Example 1: Input: nums = [5,2,3,1] Output: [1,2,3,5] Explanation: After sorting the array, the positions of some numbers are not changed (for example, 2 and 3), while the positions of other numbers are changed (for example, 1 and 5).

Example 2: Input: nums = [5,1,1,2,0,0] Output: [0,0,1,1,2,5] Explanation: Note that the values of nums are not necessairly unique.

Constraints:

1 <= nums.length <= 5 * 104 -5 * 104 <= nums[i] <= 5 * 104

==

Problem Number: 938 URL: https://leetcode.com/problems/partition-array-into-disjoint-intervals Title: 915. Partition Array into Disjoint Intervals Problem Description: Given an integer array nums, partition it into two (contiguous) subarrays left and right so that:

Every element in left is less than or equal to every element in right. left and right are non-empty. left has the smallest possible size.

Return the length of left after such a partitioning. Test cases are generated such that partitioning exists. Example 1: Input: nums = [5,0,3,8,6] Output: 3 Explanation: left = [5,0,3], right = [8,6]

Example 2: Input: nums = [1,1,1,0,6,12] Output: 4 Explanation: left = [1,1,1,0], right = [6,12]

Constraints:

2 <= nums.length <= 105 0 <= nums[i] <= 106 There is at least one valid answer for the given input.

==

Problem Number: 939 URL: https://leetcode.com/problems/word-subsets Title: 916. Word Subsets Problem Description: You are given two string arrays words1 and words2. A string b is a subset of string a if every letter in b occurs in a including multiplicity.

For example, "wrr" is a subset of "warrior" but is not a subset of "world".

A string a from words1 is universal if for every string b in words2, b is a subset of a. Return an array of all the universal strings in words1. You may return the answer in any order. Example 1: Input: words1 = ["amazon","apple","facebook","google","leetcode"], words2 = ["e","o"] Output: ["facebook","google","leetcode"]

Example 2: Input: words1 = ["amazon","apple","facebook","google","leetcode"], words2 = ["l","e"] Output: ["apple","google","leetcode"]

Constraints:

1 <= words1.length, words2.length <= 104 1 <= words1[i].length, words2[i].length <= 10 words1[i] and words2[i] consist only of lowercase English letters. All the strings of words1 are unique.

==
Problem Number: 940 URL: https://leetcode.com/problems/maximum-sum-circular-subarray Title: 918. Maximum Sum Circular Subarray Problem Description: Given a circular integer array nums of length n, return the maximum possible sum of a non-empty subarray of nums. A circular array means the end of the array connects to the beginning of the array. Formally, the next element of nums[i] is nums[(i + 1) % n] and the previous element of nums[i] is nums[(i - 1 + n) % n]. A subarray may only include each element of the fixed buffer nums at most once. Formally, for a subarray nums[i], nums[i + 1], ..., nums[j], there does not exist i <= k1, k2 <= j with k1 % n == k2 % n. Example 1: Input: nums = [1,-2,3,-2] Output: 3 Explanation: Subarray [3] has maximum sum 3.

Example 2: Input: nums = [5,-3,5] Output: 10 Explanation: Subarray [5,5] has maximum sum 5 + 5 = 10.

Example 3: Input: nums = [-3,-2,-3] Output: -2 Explanation: Subarray [-2] has maximum sum -2.

Constraints:

n == nums.length 1 <= n <= 3 * 104 -3 * 104 <= nums[i] <= 3 * 104

==
Problem Number: 941 URL: https://leetcode.com/problems/complete-binary-tree-inserter Title: 919. Complete Binary Tree Inserter Problem Description: A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. Design an algorithm to insert a new node to a complete binary tree keeping it complete after the insertion. Implement the CBTInserter class:

CBTInserter(TreeNode root) Initializes the data structure with the root of the complete binary tree. int insert(int v) Inserts a TreeNode into the tree with value Node.val == val so that the tree remains complete, and returns the value of the parent of the inserted TreeNode. TreeNode get_root() Returns the root node of the tree.

Example 1:

Input ["CBTInserter", "insert", "insert", "get_root"] [[[1, 2]], [3], [4], []] Output [null, 1, 2, [1, 2, 3, 4]]

Explanation CBTInserter cBTInserter = new CBTInserter([1, 2]); cBTInserter.insert(3); // return 1 cBTInserter.insert(4); // return 2 cBTInserter.get_root(); // return [1, 2, 3, 4]

Constraints:

The number of nodes in the tree will be in the range [1, 1000]. 0 <= Node.val <= 5000 root is a complete binary tree. 0 <= val <= 5000 At most 104 calls will be made to insert and get_root.

==
Problem Number: 942 URL: https://leetcode.com/problems/minimum-add-to-make-parentheses-valid Title: 921. Minimum Add to Make Parentheses Valid Problem Description: A parentheses string is valid if and only if:

It is the empty string, It can be written as AB (A concatenated with B), where A and B are valid strings, or It can be written as (A), where A is a valid string.

You are given a parentheses string s. In one move, you can insert a parenthesis at any position of the string.

For example, if s = "()))", you can insert an opening parenthesis to be "(()))" or a closing parenthesis to be "())))".

Return the minimum number of moves required to make s valid.   Example 1: Input: s = "())" Output: 1

Example 2: Input: s = "(((" Output: 3

Constraints:

1 <= s.length <= 1000 s[i] is either '(' or ')'.

==
Problem Number: 943 URL: https://leetcode.com/problems/3sum-with-multiplicity Title: 923. 3Sum With Multiplicity Problem Description: Given an integer array arr, and an integer target, return the number of tuples i, j, k such that i < j < k and arr[i] + arr[j] + arr[k] == target. As the answer can be very large, return it modulo 109 + 7.   Example 1: Input: arr = [1,1,2,2,3,3,4,4,5,5], target = 8 Output: 20 Explanation: Enumerating by the values (arr[i], arr[j], arr[k]): (1, 2, 5) occurs 8 times; (1, 3, 4) occurs 8 times; (2, 2, 4) occurs 2 times; (2, 3, 3) occurs 2 times.

Example 2: Input: arr = [1,1,2,2,2,2], target = 5 Output: 12 Explanation: arr[i] = 1, arr[j] = arr[k] = 2 occurs 12 times: We choose one 1 from [1,1] in 2 ways, and two 2s from [2,2,2,2] in 6 ways.

Example 3: Input: arr = [2,1,3], target = 6 Output: 1 Explanation: (1, 2, 3) occured one time in the array so we return 1.

Constraints:

3 <= arr.length <= 3000 0 <= arr[i] <= 100 0 <= target <= 300

== Problem Number: 944 URL: https://leetcode.com/problems/flip-string-to-monotone-increasing Title: 926. Flip String to Monotone Increasing Problem Description: A binary string is monotone increasing if it consists of some number of 0's (possibly none), followed by some number of 1's (also possibly none). You are given a binary string s. You can flip s[i] changing it from 0 to 1 or from 1 to 0. Return the minimum number of flips to make s monotone increasing. Example 1: Input: s = "00110" Output: 1 Explanation: We flip the last digit to get 00111.

Example 2: Input: s = "010110" Output: 2 Explanation: We flip to get 011111, or alternatively 000111.

Example 3: Input: s = "00011000" Output: 2 Explanation: We flip to get 00000000.

Constraints:

1 <= s.length <= 105 s[i] is either '0' or '1'.

== Problem Number: 945 URL: https://leetcode.com/problems/binary-subarrays-with-sum Title: 930. Binary Subarrays With Sum Problem Description: Given a binary array nums and an integer goal, return the number of non-empty subarrays with a sum goal. A subarray is a contiguous part of the array. Example 1: Input: nums = [1,0,1,0,1], goal = 2 Output: 4 Explanation: The 4 subarrays are bolded and underlined below: [1,0,1,0,1] [1,0,1,0,1] [1,0,1,0,1] [1,0,1,0,1]

Example 2: Input: nums = [0,0,0,0,0], goal = 0 Output: 15

Constraints:

1 <= nums.length <= 3 * 104 nums[i] is either 0 or 1. 0 <= goal <= nums.length

== Problem Number: 946 URL: https://leetcode.com/problems/minimum-falling-path-sum Title: 931. Minimum Falling Path Sum Problem Description: Given an n x n array of integers matrix, return the minimum sum of any falling path through matrix. A falling path starts at any element in the first row and chooses the element in the next row that is either directly below or diagonally left/right. Specifically, the next element from position (row, col) will be (row + 1, col - 1), (row + 1, col), or (row + 1, col + 1). Example 1:

Input: matrix = [[2,1,3],[6,5,4],[7,8,9]] Output: 13 Explanation: There are two falling paths with a minimum sum as shown.

Example 2:

Input: matrix = [[-19,57],[-40,-5]] Output: -59 Explanation: The falling path with a minimum sum is shown.

Constraints:

n == matrix.length == matrix[i].length 1 <= n <= 100 -100 <= matrix[i][j] <= 100

==

Problem Number: 947 URL: https://leetcode.com/problems/beautiful-array Title: 932. Beautiful Array Problem Description: An array nums of length n is beautiful if:

nums is a permutation of the integers in the range [1, n]. For every 0 <= i < j < n, there is no index k with i < k < j where 2 * nums[k] == nums[i] + nums[j].

Given the integer n, return any beautiful array nums of length n. There will be at least one valid answer for the given n. Example 1: Input: n = 4 Output: [2,1,4,3] Example 2: Input: n = 5 Output: [3,1,2,5,4]

Constraints:

1 <= n <= 1000

==

Problem Number: 948 URL: https://leetcode.com/problems/shortest-bridge Title: 934. Shortest Bridge Problem Description: You are given an n x n binary matrix grid where 1 represents land and 0 represents water. An island is a 4-directionally connected group of 1's not connected to any other 1's. There are exactly two islands in grid. You may change 0's to 1's to connect the two islands to form one island. Return the smallest number of 0's you must flip to connect the two islands. Example 1: Input: grid = [[0,1],[1,0]] Output: 1

Example 2: Input: grid = [[0,1,0],[0,0,0],[0,0,1]] Output: 2

Example 3: Input: grid = [[1,1,1,1,1],[1,0,0,0,1],[1,0,1,0,1],[1,0,0,0,1],[1,1,1,1,1]] Output: 1

Constraints:

n == grid.length == grid[i].length 2 <= n <= 100 grid[i][j] is either 0 or 1. There are exactly two islands in grid.

==

Problem Number: 949 URL: https://leetcode.com/problems/knight-dialer Title: 935. Knight Dialer Problem Description: The chess knight has a unique movement, it may move two squares vertically and one square horizontally, or two squares horizontally and one square vertically (with both forming the shape of an L). The possible movements of chess knight are shown in this diagaram: A chess knight can move as indicated in the chess diagram below:

We have a chess knight and a phone pad as shown below, the knight can only stand on a numeric cell (i.e. blue cell).

Given an integer n, return how many distinct phone numbers of length n we can dial. You are allowed to place the knight on any numeric cell initially and then you should perform n - 1 jumps to dial a number of length n. All jumps should be valid knight jumps. As the answer may be very large, return the answer modulo 109 + 7. Example 1: Input: n = 1 Output: 10 Explanation: We need to dial a number of length 1, so placing the knight over any numeric cell of the 10 cells is sufficient.

Example 2: Input: n = 2 Output: 20 Explanation: All the valid number we can dial are [04, 06, 16, 18, 27, 29, 34, 38, 40, 43, 49, 60, 61, 67, 72, 76, 81, 83, 92, 94]

Example 3: Input: n = 3131 Output: 136006598 Explanation: Please take care of the mod.

Constraints:

1 <= n <= 5000

==

Problem Number: 950 URL: https://leetcode.com/problems/reorder-data-in-log-files Title: 937. Reorder Data in Log Files Problem Description: You are given an array of logs. Each log is a space-delimited string of words, where the first word is the identifier. There are two types of logs:

Letter-logs: All words (except the identifier) consist of lowercase English letters. Digit-logs: All words (except the identifier) consist of digits.

Reorder these logs so that:

The letter-logs come before all digit-logs. The letter-logs are sorted lexicographically by their contents. If their contents are the same, then sort them lexicographically by their identifiers. The digit-logs maintain their relative ordering.

Return the final order of the logs. Example 1: Input: logs = ["dig1 8 1 5 1","let1 art can","dig2 3 6","let2 own kit dig","let3 art zero"] Output: ["let1 art can","let3 art zero","let2 own kit dig","dig1 8 1 5 1","dig2 3 6"] Explanation: The letter-log contents are all different, so their ordering is "art can", "art zero", "own kit dig". The digit-logs have a relative order of "dig1 8 1 5 1", "dig2 3 6".

Example 2: Input: logs = ["a1 9 2 3 1","g1 act car","zo4 4 7","ab1 off key dog","a8 act zoo"] Output: ["g1 act car","a8 act zoo","ab1 off key dog","a1 9 2 3 1","zo4 4 7"]

Constraints:

1 <= logs.length <= 100 3 <= logs[i].length <= 100 All the tokens of logs[i] are separated by a single space. logs[i] is guaranteed to have an identifier and at least one word after the identifier.

===
Problem Number: 951 URL: https://leetcode.com/problems/minimum-area-rectangle Title: 939. Minimum Area Rectangle Problem Description: You are given an array of points in the X-Y plane points where points[i] = [xi, yi]. Return the minimum area of a rectangle formed from these points, with sides parallel to the X and Y axes. If there is not any such rectangle, return 0. Example 1:

Input: points = [[1,1],[1,3],[3,1],[3,3],[2,2]] Output: 4

Example 2:

Input: points = [[1,1],[1,3],[3,1],[3,3],[4,1],[4,3]] Output: 2

Constraints:

1 <= points.length <= 500 points[i].length == 2 0 <= xi, yi <= 4 * 104 All the given points are unique.

===
Problem Number: 952 URL: https://leetcode.com/problems/minimum-increment-to-make-array-unique Title: 945. Minimum Increment to Make Array Unique Problem Description: You are given an integer array nums. In one move, you can pick an index i where 0 <= i < nums.length and increment nums[i] by 1. Return the minimum number of moves to make every value in nums unique. The test cases are generated so that the answer fits in a 32-bit integer. Example 1: Input: nums = [1,2,2] Output: 1 Explanation: After 1 move, the array could be [1, 2, 3].

Example 2: Input: nums = [3,2,1,2,1,7] Output: 6 Explanation: After 6 moves, the array could be [3, 4, 1, 2, 5, 7]. It can be shown with 5 or less moves that it is impossible for the array to have all unique values.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 105

===
Problem Number: 953 URL: https://leetcode.com/problems/validate-stack-sequences Title: 946. Validate Stack Sequences Problem Description: Given two integer arrays pushed and popped each with distinct values, return true if this could have been the result of a sequence of push and pop operations on an initially empty stack, or false otherwise. Example 1: Input: pushed = [1,2,3,4,5], popped = [4,5,3,2,1] Output: true Explanation: We might do the following sequence: push(1), push(2), push(3), push(4), pop() -> 4, push(5), pop() -> 5, pop() -> 3, pop() -> 2, pop() -> 1

Example 2: Input: pushed = [1,2,3,4,5], popped = [4,3,5,1,2] Output: false Explanation: 1 cannot be popped before 2.

Constraints:

1 <= pushed.length <= 1000 0 <= pushed[i] <= 1000 All the elements of pushed are unique. popped.length == pushed.length popped is a permutation of pushed.

==

Problem Number: 954 URL: https://leetcode.com/problems/most-stones-removed-with-same-row-or-column Title: 947. Most Stones Removed with Same Row or Column Problem Description: On a 2D plane, we place n stones at some integer coordinate points. Each coordinate point may have at most one stone. A stone can be removed if it shares either the same row or the same column as another stone that has not been removed. Given an array stones of length n where stones[i] = [xi, yi] represents the location of the ith stone, return the largest possible number of stones that can be removed. Example 1: Input: stones = [[0,0],[0,1],[1,0],[1,2],[2,1],[2,2]] Output: 5 Explanation: One way to remove 5 stones is as follows: 1. Remove stone [2,2] because it shares the same row as [2,1]. 2. Remove stone [2,1] because it shares the same column as [0,1]. 3. Remove stone [1,2] because it shares the same row as [1,0]. 4. Remove stone [1,0] because it shares the same column as [0,0]. 5. Remove stone [0,1] because it shares the same row as [0,0]. Stone [0,0] cannot be removed since it does not share a row/column with another stone still on the plane.

Example 2: Input: stones = [[0,0],[0,2],[1,1],[2,0],[2,2]] Output: 3 Explanation: One way to make 3 moves is as follows: 1. Remove stone [2,2] because it shares the same row as [2,0]. 2. Remove stone [2,0] because it shares the same column as [0,0]. 3. Remove stone [0,2] because it shares the same row as [0,0]. Stones [0,0] and [1,1] cannot be removed since they do not share a row/column with another stone still on the plane.

Example 3: Input: stones = [[0,0]] Output: 0 Explanation: [0,0] is the only stone on the plane, so you cannot remove it.

Constraints:

1 <= stones.length <= 1000 0 <= xi, yi <= 104 No two stones are at the same coordinate point.

==

Problem Number: 955 URL: https://leetcode.com/problems/bag-of-tokens Title: 948. Bag of Tokens Problem Description: You have an initial power of power, an initial score of 0, and a bag of tokens where tokens[i] is the value of the ith token (0-indexed). Your goal is to maximize your total score by potentially playing each token in one of two ways:

If your current power is at least tokens[i], you may play the ith token face up, losing tokens[i] power and gaining 1 score. If your current score is at least 1, you may play the ith token face down, gaining tokens[i] power and losing 1 score.

Each token may be played at most once and in any order. You do not have to play all the tokens. Return the largest possible score you can achieve after playing any number of tokens. Example 1: Input: tokens = [100], power = 50

Output: 0 Explanation: Playing the only token in the bag is impossible because you either have too little power or too little score.

Example 2: Input: tokens = [100,200], power = 150 Output: 1 Explanation: Play the 0th token (100) face up, your power becomes 50 and score becomes 1. There is no need to play the 1st token since you cannot play it face up to add to your score.

Example 3: Input: tokens = [100,200,300,400], power = 200 Output: 2 Explanation: Play the tokens in this order to get a score of 2: 1. Play the 0th token (100) face up, your power becomes 100 and score becomes 1. 2. Play the 3rd token (400) face down, your power becomes 500 and score becomes 0. 3. Play the 1st token (200) face up, your power becomes 300 and score becomes 1. 4. Play the 2nd token (300) face up, your power becomes 0 and score becomes 2.

Constraints:

0 <= tokens.length <= 1000 0 <= tokens[i], power < 104

==
Problem Number: 956 URL: https://leetcode.com/problems/largest-time-for-given-digits Title: 949. Largest Time for Given Digits Problem Description: Given an array arr of 4 digits, find the latest 24-hour time that can be made using each digit exactly once. 24-hour times are formatted as "HH:MM", where HH is between 00 and 23, and MM is between 00 and 59. The earliest 24-hour time is 00:00, and the latest is 23:59. Return the latest 24-hour time in "HH:MM" format. If no valid time can be made, return an empty string. Example 1: Input: arr = [1,2,3,4] Output: "23:41" Explanation: The valid 24-hour times are "12:34", "12:43", "13:24", "13:42", "14:23", "14:32", "21:34", "21:43", "23:14", and "23:41". Of these times, "23:41" is the latest.

Example 2: Input: arr = [5,5,5,5] Output: "" Explanation: There are no valid 24-hour times as "55:55" is not valid.

Constraints:

arr.length == 4 0 <= arr[i] <= 9

==
Problem Number: 957 URL: https://leetcode.com/problems/reveal-cards-in-increasing-order Title: 950. Reveal Cards In Increasing Order Problem Description: You are given an integer array deck. There is a deck of cards where every card has a unique integer. The integer on the ith card is deck[i]. You can order the deck in any order you want. Initially, all the cards start face down (unrevealed) in one deck. You will do the following steps repeatedly until all cards are revealed:

Take the top card of the deck, reveal it, and take it out of the deck. If there are still cards in the deck then put the next top card of the deck at the bottom of the deck. If there are still unrevealed cards, go back to step 1. Otherwise, stop.

Return an ordering of the deck that would reveal the cards in increasing order. Note that the first entry in the answer is considered to be the top of the deck. Example 1: Input: deck = [17,13,11,2,3,5,7] Output: [2,13,3,11,5,17,7] Explanation: We get the deck in the order [17,13,11,2,3,5,7] (this order does not matter), and reorder it. After reordering, the deck starts as [2,13,3,11,5,17,7], where 2 is the top of the deck. We reveal 2, and move 13 to the bottom. The deck is now [3,11,5,17,7,13]. We reveal 3, and move 11 to the bottom. The deck is now [5,17,7,13,11]. We reveal 5, and move 17 to the bottom. The deck is now [7,13,11,17]. We reveal 7, and move 13 to the bottom. The deck is now [11,17,13]. We reveal 11, and move 17 to the bottom. The deck is now [13,17]. We reveal 13, and move 17 to the bottom. The deck is now [17]. We reveal 17. Since all the cards revealed are in increasing order, the answer is correct.

Example 2: Input: deck = [1,1000] Output: [1,1000]

Constraints:

1 <= deck.length <= 1000 1 <= deck[i] <= 106 All the values of deck are unique.

==
Problem Number: 958 URL: https://leetcode.com/problems/flip-equivalent-binary-trees Title: 951. Flip Equivalent Binary Trees Problem Description: For a binary tree T, we can define a flip operation as follows: choose any node, and swap the left and right child subtrees. A binary tree X is flip equivalent to a binary tree Y if and only if we can make X equal to Y after some number of flip operations. Given the roots of two binary trees root1 and root2, return true if the two trees are flip equivalent or false otherwise. Example 1:

Input: root1 = [1,2,3,4,5,6,null,null,null,7,8], root2 = [1,3,2,null,6,4,5,null,null,null,null,8,7] Output: true Explanation: We flipped at nodes with values 1, 3, and 5.

Example 2: Input: root1 = [], root2 = [] Output: true

Example 3: Input: root1 = [], root2 = [1] Output: false

Constraints:

The number of nodes in each tree is in the range [0, 100]. Each tree will have unique node values in the range [0, 99].

==
Problem Number: 959 URL: https://leetcode.com/problems/array-of-doubled-pairs Title: 954. Array of Doubled Pairs Problem Description: Given an integer array of even length arr, return true if it is possible to reorder arr such that arr[2 * i + 1] = 2 * arr[2 * i] for every 0 <= i < len(arr) / 2, or false otherwise. Example 1: Input: arr = [3,1,3,6] Output: false

Example 2: Input: arr = [2,1,2,6] Output: false

Example 3: Input: arr = [4,-2,2,-4] Output: true Explanation: We can take two groups, [-2,-4] and [2,4] to form [-2,-4,2,4] or [2,4,-2,-4].

Constraints:

2 <= arr.length <= 3 * 104 arr.length is even. -105 <= arr[i] <= 105

==
Problem Number: 960 URL: https://leetcode.com/problems/delete-columns-to-make-sorted-ii Title: 955. Delete Columns to Make Sorted II Problem Description: You are given an array of n strings strs, all of the same length. We may choose any deletion indices, and we delete all the characters in those indices for each string. For example, if we have strs = ["abcdef","uvwxyz"] and deletion indices {0, 2, 3}, then the final array after deletions is ["bef", "vyz"]. Suppose we chose a set of deletion indices answer such that after deletions, the final array has its elements in lexicographic order (i.e., strs[0] <= strs[1] <= strs[2] <= ... <= strs[n - 1]). Return the minimum possible value of answer.length. Example 1: Input: strs = ["ca","bb","ac"] Output: 1 Explanation: After deleting the first column, strs = ["a", "b", "c"]. Now strs is in lexicographic order (ie. strs[0] <= strs[1] <= strs[2]). We require at least 1 deletion since initially strs was not in lexicographic order, so the answer is 1.

Example 2: Input: strs = ["xc","yb","za"] Output: 0 Explanation: strs is already in lexicographic order, so we do not need to delete anything. Note that the rows of strs are not necessarily in lexicographic order: i.e., it is NOT necessarily true that (strs[0][0] <= strs[0][1] <= ...)

Example 3: Input: strs = ["zyx","wvu","tsr"] Output: 3 Explanation: We have to delete every column.

Constraints:

n == strs.length 1 <= n <= 100 1 <= strs[i].length <= 100 strs[i] consists of lowercase English letters.

==
Problem Number: 961 URL: https://leetcode.com/problems/prison-cells-after-n-days Title: 957. Prison Cells After N Days Problem Description: There are 8 prison cells in a row and each cell is either occupied or vacant. Each day, whether the cell is occupied or vacant changes according to the following rules:

If a cell has two adjacent neighbors that are both occupied or both vacant, then the cell becomes occupied. Otherwise, it becomes vacant.

Note that because the prison is a row, the first and the last cells in the row can't have two adjacent neighbors. You are given an integer array cells where cells[i] == 1 if the ith cell is occupied and cells[i] == 0 if the ith cell is vacant, and you are given an integer n. Return the state of the prison after n days (i.e., n such changes described above). Example 1: Input: cells = [0,1,0,1,1,0,0,1], n = 7 Output: [0,0,1,1,0,0,0,0] Explanation: The following table summarizes the state of the prison on each day: Day 0: [0, 1, 0, 1, 1, 0, 0, 1] Day 1: [0, 1, 1, 0, 0, 0, 0, 0] Day 2: [0, 0, 0, 0, 1, 1, 1, 0] Day 3: [0, 1, 1, 0, 0, 1, 0, 0] Day 4: [0, 0, 0, 0, 0, 1, 0, 0] Day 5: [0, 1, 1, 1, 0, 1, 0, 0] Day 6: [0, 0, 1, 0, 1, 1, 0, 0] Day 7: [0, 0, 1, 1, 0, 0, 0, 0]

Example 2: Input: cells = [1,0,0,1,0,0,1,0], n = 1000000000 Output: [0,0,1,1,1,1,1,0]

Constraints:

cells.length == 8 cells[i] is either 0 or 1. 1 <= n <= 109

==
Problem Number: 962 URL: https://leetcode.com/problems/check-completeness-of-a-binary-tree Title: 958. Check Completeness of a Binary Tree Problem Description: Given the root of a binary tree, determine if it is a complete binary tree. In a complete binary tree, every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible. It can have between 1 and 2h nodes inclusive at the last level h. Example 1:

Input: root = [1,2,3,4,5,6] Output: true Explanation: Every level before the last is full (ie. levels with node-values {1} and {2, 3}), and all nodes in the last level ({4, 5, 6}) are as far left as possible.

Example 2:

Input: root = [1,2,3,4,5,null,7] Output: false Explanation: The node with value 7 isn't as far left as possible.

Constraints:

The number of nodes in the tree is in the range [1, 100]. 1 <= Node.val <= 1000

==
Problem Number: 963 URL: https://leetcode.com/problems/regions-cut-by-slashes Title: 959. Regions Cut By Slashes Problem Description: An n x n grid is composed of 1 x 1 squares where each 1 x 1 square consists of a '/', '\', or blank space ' '. These characters divide the square into contiguous regions. Given the grid grid represented as a string array, return the number of regions. Note that backslash characters are escaped, so a '\' is represented as '\\'. Example 1:

Input: grid = [" /","/ "] Output: 2

Example 2:

Input: grid = [" /"," "] Output: 1

Example 3:

Input: grid = ["/\\","\\/"] Output: 5 Explanation: Recall that because \ characters are escaped, "\\/" refers to \/, and "/\\" refers to /\.

Constraints:

n == grid.length == grid[i].length 1 <= n <= 30 grid[i][j] is either '/', '\', or ' '.

==

Problem Number: 964 URL: https://leetcode.com/problems/maximum-width-ramp Title: 962. Maximum Width Ramp Problem Description: A ramp in an integer array nums is a pair (i, j) for which i < j and nums[i] <= nums[j]. The width of such a ramp is j - i. Given an integer array nums, return the maximum width of a ramp in nums. If there is no ramp in nums, return 0. Example 1: Input: nums = [6,0,8,2,1,5] Output: 4 Explanation: The maximum width ramp is achieved at (i, j) = (1, 5): nums[1] = 0 and nums[5] = 5.

Example 2: Input: nums = [9,8,1,0,1,9,4,0,4,1] Output: 7 Explanation: The maximum width ramp is achieved at (i, j) = (2, 9): nums[2] = 1 and nums[9] = 1.

Constraints:

2 <= nums.length <= 5 * 104 0 <= nums[i] <= 5 * 104

==

Problem Number: 965 URL: https://leetcode.com/problems/minimum-area-rectangle-ii Title: 963. Minimum Area Rectangle II Problem Description: You are given an array of points in the X-Y plane points where points[i] = [xi, yi]. Return the minimum area of any rectangle formed from these points, with sides not necessarily parallel to the X and Y axes. If there is not any such rectangle, return 0. Answers within 10-5 of the actual answer will be accepted. Example 1:

Input: points = [[1,2],[2,1],[1,0],[0,1]] Output: 2.00000 Explanation: The minimum area rectangle occurs at [1,2],[2,1],[1,0],[0,1], with an area of 2.

Example 2:

Input: points = [[0,1],[2,1],[1,1],[1,0],[2,0]] Output: 1.00000 Explanation: The minimum area rectangle occurs at [1,0],[1,1],[2,1],[2,0], with an area of 1.

Example 3:

Input: points = [[0,3],[1,2],[3,1],[1,3],[2,1]] Output: 0 Explanation: There is no possible rectangle to form from these points.

Constraints:

1 <= points.length <= 50 points[i].length == 2 0 <= xi, yi <= 4 * 104 All the given points are unique.

==

Problem Number: 966 URL: https://leetcode.com/problems/vowel-spellchecker Title: 966. Vowel Spellchecker Problem Description: Given a wordlist, we want to implement a spellchecker that converts a query word into a correct word.

For a given query word, the spell checker handles two categories of spelling mistakes:

Capitalization: If the query matches a word in the wordlist (case-insensitive), then the query word is returned with the same case as the case in the wordlist.

Example: wordlist = ["yellow"], query = "YellOw": correct = "yellow" Example: wordlist = ["Yellow"], query = "yellow": correct = "Yellow" Example: wordlist = ["yellow"], query = "yellow": correct = "yellow"

Vowel Errors: If after replacing the vowels ('a', 'e', 'i', 'o', 'u') of the query word with any vowel individually, it matches a word in the wordlist (case-insensitive), then the query word is returned with the same case as the match in the wordlist.

Example: wordlist = ["YellOw"], query = "yollow": correct = "YellOw" Example: wordlist = ["YellOw"], query = "yeellow": correct = "" (no match) Example: wordlist = ["YellOw"], query = "yllw": correct = "" (no match)

In addition, the spell checker operates under the following precedence rules:

When the query exactly matches a word in the wordlist (case-sensitive), you should return the same word back. When the query matches a word up to capitlization, you should return the first such match in the wordlist. When the query matches a word up to vowel errors, you should return the first such match in the wordlist. If the query has no matches in the wordlist, you should return the empty string.

Given some queries, return a list of words answer, where answer[i] is the correct word for query = queries[i]. Example 1: Input: wordlist = ["KiTe","kite","hare","Hare"], queries = ["kite","Kite","KiTe","Hare","HARE","Hear","hear","keti","keet","k Output: ["kite","KiTe","KiTe","Hare","hare","","","KiTe","","KiTe"] Example 2: Input: wordlist = ["yellow"], queries = ["YellOw"] Output: ["yellow"]

Constraints:

1 <= wordlist.length, queries.length <= 5000 1 <= wordlist[i].length, queries[i].length <= 7 wordlist[i] and queries[i] consist only of only English letters.

== Problem Number: 967 URL: https://leetcode.com/problems/numbers-with-same-consecutive-differences Title: 967. Numbers With Same Consecutive Differences Problem Description: Given two integers n and k, return an array of all the integers of length n where the difference between every two consecutive digits is k. You may return the answer in any order. Note that the integers should not have leading zeros. Integers as 02 and 043 are not allowed. Example 1: Input: n = 3, k = 7 Output: [181,292,707,818,929] Explanation: Note that 070 is not a valid number, because it has leading zeroes.

Example 2: Input: n = 2, k = 1 Output: [10,12,21,23,32,34,43,45,54,56,65,67,76,78,87,89,98]

Constraints:

2 <= n <= 9 0 <= k <= 9

==
Problem Number: 968 URL: https://leetcode.com/problems/pancake-sorting Title: 969. Pancake Sorting Problem Description: Given an array of integers arr, sort the array by performing a series of pancake flips. In one pancake flip we do the following steps:

Choose an integer k where 1 <= k <= arr.length. Reverse the sub-array arr[0...k-1] (0-indexed).

For example, if arr = [3,2,1,4] and we performed a pancake flip choosing k = 3, we reverse the sub-array [3,2,1], so arr = [1,2,3,4] after the pancake flip at k = 3. Return an array of the k-values corresponding to a sequence of pancake flips that sort arr. Any valid answer that sorts the array within 10 * arr.length flips will be judged as correct. Example 1: Input: arr = [3,2,4,1] Output: [4,2,4,3] Explanation: We perform 4 pancake flips, with k values 4, 2, 4, and 3. Starting state: arr = [3, 2, 4, 1] After 1st flip (k = 4): arr = [1, 4, 2, 3] After 2nd flip (k = 2): arr = [4, 1, 2, 3] After 3rd flip (k = 4): arr = [3, 2, 1, 4] After 4th flip (k = 3): arr = [1, 2, 3, 4], which is sorted.

Example 2: Input: arr = [1,2,3] Output: [] Explanation: The input is already sorted, so there is no need to flip anything. Note that other answers, such as [3, 3], would also be accepted.

Constraints:

1 <= arr.length <= 100 1 <= arr[i] <= arr.length All integers in arr are unique (i.e. arr is a permutation of the integers from 1 to arr.length).

==
Problem Number: 969 URL: https://leetcode.com/problems/powerful-integers Title: 970. Powerful Integers Problem Description: Given three integers x, y, and bound, return a list of all the powerful integers that have a value less than or equal to bound. An integer is powerful if it can be represented as xi + yj for some integers i >= 0 and j >= 0. You may return the answer in any order. In your answer, each value should occur at most once. Example 1: Input: x = 2, y = 3, bound = 10 Output: [2,3,4,5,7,9,10] Explanation: 2 = 20 + 30 3 = 21 + 30 4 = 20 + 31 5 = 21 + 31 7 = 22 + 31 9 = 23 + 30 10 = 20 + 32

Example 2: Input: x = 3, y = 5, bound = 15 Output: [2,4,6,8,10,14]

Constraints:

1 <= x, y <= 100 0 <= bound <= 106

==
Problem Number: 970 URL: https://leetcode.com/problems/flip-binary-tree-to-match-preorder-traversal Title: 971. Flip Binary Tree To Match Preorder Traversal Problem Description: You are given the root of a binary tree with n nodes, where each node is uniquely assigned a value from 1 to n. You are also

given a sequence of n values voyage, which is the desired pre-order traversal of the binary tree. Any node in the binary tree can be flipped by swapping its left and right subtrees. For example, flipping node 1 will have the following effect:

Flip the smallest number of nodes so that the pre-order traversal of the tree matches voyage. Return a list of the values of all flipped nodes. You may return the answer in any order. If it is impossible to flip the nodes in the tree to make the pre-order traversal match voyage, return the list [-1]. Example 1:

Input: root = [1,2], voyage = [2,1] Output: [-1] Explanation: It is impossible to flip the nodes such that the pre-order traversal matches voyage.

Example 2:

Input: root = [1,2,3], voyage = [1,3,2] Output: [1] Explanation: Flipping node 1 swaps nodes 2 and 3, so the pre-order traversal matches voyage. Example 3:

Input: root = [1,2,3], voyage = [1,2,3] Output: [] Explanation: The tree's pre-order traversal already matches voyage, so no nodes need to be flipped.

Constraints:

The number of nodes in the tree is n. n == voyage.length 1 <= n <= 100 1 <= Node.val, voyage[i] <= n All the values in the tree are unique. All the values in voyage are unique.

==

Problem Number: 971 URL: https://leetcode.com/problems/k-closest-points-to-origin Title: 973. K Closest Points to Origin Problem Description: Given an array of points where points[i] = [xi, yi] represents a point on the X-Y plane and an integer k, return the k closest points to the origin (0, 0). The distance between two points on the X-Y plane is the Euclidean distance (i.e., √(x1 - x2)2 + (y1 - y2)2). You may return the answer in any order. The answer is guaranteed to be unique (except for the order that it is in). Example 1:

Input: points = [[1,3],[-2,2]], k = 1 Output: [[-2,2]] Explanation: The distance between (1, 3) and the origin is sqrt(10). The distance between (-2, 2) and the origin is sqrt(8). Since sqrt(8) < sqrt(10), (-2, 2) is closer to the origin. We only want the closest k = 1 points from the origin, so the answer is just [[-2,2]].

Example 2: Input: points = [[3,3],[5,-1],[-2,4]], k = 2 Output: [[3,3],[-2,4]] Explanation: The answer [[-2,4],[3,3]] would also be accepted.

Constraints:

1 <= k <= points.length <= 104 -104 <= xi, yi <= 104

==

Problem Number: 972 URL: https://leetcode.com/problems/subarray-sums-divisible-by-k Title: 974. Subarray Sums Divisible by K Problem Description: Given an integer array nums and an integer k, return the number of non-empty subarrays that have a sum divisible by k. A subarray is a contiguous part of an array. Example 1: Input: nums = [4,5,0,-2,-3,1], k = 5 Output: 7 Explanation: There are 7 subarrays with a sum divisible by k = 5: [4, 5, 0, -2, -3, 1], [5], [5, 0], [5, 0, -2, -3], [0], [0, -2, -3], [-2, -3]

Example 2: Input: nums = [5], k = 9 Output: 0

Constraints:

1 <= nums.length <= 3 * 104 -104 <= nums[i] <= 104 2 <= k <= 104

==
Problem Number: 973 URL: https://leetcode.com/problems/longest-turbulent-subarray Title: 978. Longest Turbulent Subarray Problem Description: Given an integer array arr, return the length of a maximum size turbulent subarray of arr. A subarray is turbulent if the comparison sign flips between each adjacent pair of elements in the subarray. More formally, a subarray [arr[i], arr[i + 1], ..., arr[j]] of arr is said to be turbulent if and only if:

For i <= k < j:

arr[k] > arr[k + 1] when k is odd, and arr[k] < arr[k + 1] when k is even.

Or, for i <= k < j:

arr[k] > arr[k + 1] when k is even, and arr[k] < arr[k + 1] when k is odd.

Example 1: Input: arr = [9,4,2,10,7,8,8,1,9] Output: 5 Explanation: arr[1] > arr[2] < arr[3] > arr[4] < arr[5]

Example 2: Input: arr = [4,8,12,16] Output: 2

Example 3: Input: arr = [100] Output: 1

Constraints:

1 <= arr.length <= 4 * 104 0 <= arr[i] <= 109

==
Problem Number: 974 URL: https://leetcode.com/problems/distribute-coins-in-binary-tree Title: 979. Distribute Coins in Binary Tree Problem Description: You are given the root of a binary tree with n nodes where each node in the tree has node.val coins. There are n coins in total throughout the whole tree. In one move, we may choose two adjacent nodes and move one coin from one node to another. A move may be from parent to child, or from child to parent. Return the minimum number of moves required to make every node have exactly one coin. Example 1:

Input: root = [3,0,0] Output: 2 Explanation: From the root of the tree, we move one coin to its left child, and one coin to its right child.

Example 2:

Input: root = [0,3,0] Output: 3 Explanation: From the left child of the root, we move two coins to the root [taking two moves]. Then, we move one coin from the root of the tree to the right child.

Constraints:

The number of nodes in the tree is n. 1 <= n <= 100 0 <= Node.val <= n The sum of all Node.val is n.

==
Problem Number: 975 URL: https://leetcode.com/problems/time-based-key-value-store Title: 981. Time Based Key-Value Store Problem Description: Design a time-based key-value data structure that can store multiple values for the same key at different time stamps and retrieve the key's value at a certain timestamp. Implement the TimeMap class:

TimeMap() Initializes the object of the data structure. void set(String key, String value, int timestamp) Stores the key key with the value value at the given time timestamp. String get(String key, int timestamp) Returns a value such that set was called previously, with timestamp_prev <= timestamp. If there are multiple such values, it returns the value associated with the largest timestamp_prev. If there are no values, it returns "".

Example 1: Input ["TimeMap", "set", "get", "get", "set", "get", "get"] [[], ["foo", "bar", 1], ["foo", 1], ["foo", 3], ["foo", "bar2", 4], ["foo", 4], ["foo", 5]] Output [null, null, "bar", "bar", null, "bar2", "bar2"]

Explanation TimeMap timeMap = new TimeMap(); timeMap.set("foo", "bar", 1); // store the key "foo" and value "bar" along with timestamp = 1. timeMap.get("foo", 1); // return "bar" timeMap.get("foo", 3); // return "bar", since there is no value corresponding to foo at timestamp 3 and timestamp 2, then the only value is at timestamp 1 is "bar". timeMap.set("foo", "bar2", 4); // store the key "foo" and value "bar2" along with timestamp = 4. timeMap.get("foo", 4); // return "bar2" timeMap.get("foo", 5); // return "bar2"

Constraints:

1 <= key.length, value.length <= 100 key and value consist of lowercase English letters and digits. 1 <= timestamp <= 107 All the timestamps timestamp of set are strictly increasing. At most 2 * 105 calls will be made to set and get.

==
Problem Number: 976 URL: https://leetcode.com/problems/minimum-cost-for-tickets Title: 983. Minimum Cost For Tickets Problem Description: You have planned some train traveling one year in advance. The days of the year in which you will travel are given as an integer array days. Each day is an integer from 1 to 365. Train tickets are sold in three different ways:

a 1-day pass is sold for costs[0] dollars, a 7-day pass is sold for costs[1] dollars, and a 30-day pass is sold for costs[2] dollars.

The passes allow that many days of consecutive travel.

For example, if we get a 7-day pass on day 2, then we can travel for 7 days: 2, 3, 4, 5, 6, 7, and 8.

Return the minimum number of dollars you need to travel every day in the given list of days. Example 1: Input: days = [1,4,6,7,8,20], costs = [2,7,15] Output: 11 Explanation: For example, here is one way to buy passes that lets you travel your travel plan: On day 1, you bought a 1-day pass for costs[0] = $2, which covered day 1. On day 3, you bought a 7-day pass for costs[1] = $7, which covered days 3, 4, ..., 9. On day 20, you bought a 1-day pass for costs[0] = $2, which covered day 20. In total, you spent $11 and covered all the days of your travel.

Example 2: Input: days = [1,2,3,4,5,6,7,8,9,10,30,31], costs = [2,7,15] Output: 17 Explanation: For example, here is one way to buy passes that lets you travel your travel plan: On day 1, you bought a 30-day pass for costs[2] = $15 which covered days 1, 2, ..., 30. On day 31, you bought a 1-day pass for costs[0] = $2 which covered day 31. In total, you spent $17 and covered all the days of your travel.

Constraints:

1 <= days.length <= 365 1 <= days[i] <= 365 days is in strictly increasing order. costs.length == 3 1 <= costs[i] <= 1000

==
Problem Number: 977 URL: https://leetcode.com/problems/string-without-aaa-or-bbb Title: 984. String Without AAA or BBB Problem Description: Given two integers a and b, return any string s such that:

s has length a + b and contains exactly a 'a' letters, and exactly b 'b' letters, The substring 'aaa' does not occur in s, and The substring 'bbb' does not occur in s.

Example 1: Input: a = 1, b = 2 Output: "abb" Explanation: "abb", "bab" and "bba" are all correct answers.

Example 2: Input: a = 4, b = 1 Output: "aabaa"

Constraints:

0 <= a, b <= 100 It is guaranteed such an s exists for the given a and b.

==
Problem Number: 978 URL: https://leetcode.com/problems/sum-of-even-numbers-after-queries Title: 985. Sum of Even Numbers After Queries Problem Description: You are given an integer array nums and an array queries where queries[i] = [vali, indexi]. For each query i, first, apply nums[indexi] = nums[indexi] + vali, then print the sum of the even values of nums. Return an integer array answer where answer[i] is the answer to the ith query. Example 1: Input: nums = [1,2,3,4], queries = [[1,0],[-3,1],[-4,0],[2,3]] Output: [8,6,2,4]

Explanation: At the beginning, the array is [1,2,3,4]. After adding 1 to nums[0], the array is [2,2,3,4], and the sum of even values is 2 + 2 + 4 = 8. After adding -3 to nums[1], the array is [2,-1,3,4], and the sum of even values is 2 + 4 = 6. After adding -4 to nums[0], the array is [-2,-1,3,4], and the sum of even values is -2 + 4 = 2. After adding 2 to nums[3], the array is [-2,-1,3,6], and the sum of even values is -2 + 6 = 4.

Example 2: Input: nums = [1], queries = [[4,0]] Output: [0]

Constraints:

1 <= nums.length <= 104 -104 <= nums[i] <= 104 1 <= queries.length <= 104 -104 <= vali <= 104 0 <= indexi < nums.length

==
Problem Number: 979 URL: https://leetcode.com/problems/interval-list-intersections Title: 986. Interval List Intersections Problem Description: You are given two lists of closed intervals, firstList and secondList, where firstList[i] = [starti, endi] and secondList[j] = [startj, endj]. Each list of intervals is pairwise disjoint and in sorted order. Return the intersection of these two interval lists. A closed interval [a, b] (with a <= b) denotes the set of real numbers x with a <= x <= b. The intersection of two closed intervals is a set of real numbers that are either empty or represented as a closed interval. For example, the intersection of [1, 3] and [2, 4] is [2, 3]. Example 1:

Input: firstList = [[0,2],[5,10],[13,23],[24,25]], secondList = [[1,5],[8,12],[15,24],[25,26]] Output: [[1,2],[5,5],[8,10],[15,23],[24,24],[25,25]]

Example 2: Input: firstList = [[1,3],[5,9]], secondList = [] Output: []

Constraints:

0 <= firstList.length, secondList.length <= 1000 firstList.length + secondList.length >= 1 0 <= starti < endi <= 109 endi < starti+1 0 <= startj < endj <= 109 endj < startj+1

==
Problem Number: 980 URL: https://leetcode.com/problems/smallest-string-starting-from-leaf Title: 988. Smallest String Starting From Leaf Problem Description: You are given the root of a binary tree where each node has a value in the range [0, 25] representing the letters 'a' to 'z'. Return the lexicographically smallest string that starts at a leaf of this tree and ends at the root. As a reminder, any shorter prefix of a string is lexicographically smaller.

For example, "ab" is lexicographically smaller than "aba".

A leaf of a node is a node that has no children. Example 1:

Input: root = [0,1,2,3,4,3,4] Output: "dba"

Example 2:

Input: root = [25,1,3,1,3,0,2] Output: "adz"

Example 3:

Input: root = [2,2,1,null,1,0,null,0] Output: "abc"

Constraints:

The number of nodes in the tree is in the range [1, 8500]. 0 <= Node.val <= 25

==

Problem Number: 981 URL: https://leetcode.com/problems/satisfiability-of-equality-equations Title: 990. Satisfiability of Equality Equations Problem Description: You are given an array of strings equations that represent relationships between variables where each string equations[i] is of length 4 and takes one of two different forms: "xi==yi" or "xi!=yi".Here, xi and yi are lowercase letters (not necessarily different) that represent one-letter variable names. Return true if it is possible to assign integers to variable names so as to satisfy all the given equations, or false otherwise. Example 1: Input: equations = ["a==b","b!=a"] Output: false Explanation: If we assign say, a = 1 and b = 1, then the first equation is satisfied, but not the second. There is no way to assign the variables to satisfy both equations.

Example 2: Input: equations = ["b==a","a==b"] Output: true Explanation: We could assign a = 1 and b = 1 to satisfy both equations.

Constraints:

1 <= equations.length <= 500 equations[i].length == 4 equations[i][0] is a lowercase letter. equations[i][1] is either '=' or '!'. equations[i][2] is '='. equations[i][3] is a lowercase letter.

==

Problem Number: 982 URL: https://leetcode.com/problems/broken-calculator Title: 991. Broken Calculator Problem Description: There is a broken calculator that has the integer startValue on its display initially. In one operation, you can:

multiply the number on display by 2, or subtract 1 from the number on display.

Given two integers startValue and target, return the minimum number of operations needed to display target on the calculator. Example 1: Input: startValue = 2, target = 3 Output: 2 Explanation: Use double operation and then decrement operation {2 -> 4 -> 3}.

Example 2: Input: startValue = 5, target = 8 Output: 2 Explanation: Use decrement and then double {5 -> 4 -> 8}.

Example 3: Input: startValue = 3, target = 10 Output: 3 Explanation: Use double, decrement and double {3 -> 6 -> 5 -> 10}.

Constraints:

1 <= startValue, target <= 109

==
Problem Number: 983 URL: https://leetcode.com/problems/rotting-oranges Title: 994. Rotting Oranges Problem Description: You are given an m x n grid where each cell can have one of three values:

0 representing an empty cell, 1 representing a fresh orange, or 2 representing a rotten orange.

Every minute, any fresh orange that is 4-directionally adjacent to a rotten orange becomes rotten. Return the minimum number of minutes that must elapse until no cell has a fresh orange. If this is impossible, return -1.   Example 1:

Input: grid = [[2,1,1],[1,1,0],[0,1,1]] Output: 4

Example 2: Input: grid = [[2,1,1],[0,1,1],[1,0,1]] Output: -1 Explanation: The orange in the bottom left corner (row 2, column 0) is never rotten, because rotting only happens 4-directionally.

Example 3: Input: grid = [[0,2]] Output: 0 Explanation: Since there are already no fresh oranges at minute 0, the answer is just 0.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 10 grid[i][j] is 0, 1, or 2.

==
Problem Number: 984 URL: https://leetcode.com/problems/maximum-binary-tree-ii Title: 998. Maximum Binary Tree II Problem Description: A maximum tree is a tree where every node has a value greater than any other value in its subtree. You are given the root of a maximum binary tree and an integer val. Just as in the previous problem, the given tree was constructed from a list a (root = Construct(a)) recursively with the following Construct(a) routine:

If a is empty, return null. Otherwise, let a[i] be the largest element of a. Create a root node with the value a[i]. The left child of root will be Construct([a[0], a[1], ..., a[i - 1]]). The right child of root will be Construct([a[i + 1], a[i + 2], ..., a[a.length - 1]]). Return root.

Note that we were not given a directly, only a root node root = Construct(a). Suppose b is a copy of a with the value val appended to it. It is guaranteed that b has unique values. Return Construct(b).   Example 1:

Input: root = [4,1,3,null,null,2], val = 5 Output: [5,4,null,1,3,null,null,2] Explanation: a = [1,4,2,3], b = [1,4,2,3,5]

Example 2:

Input: root = [5,2,4,null,1], val = 3 Output: [5,2,4,null,1,null,3] Explanation: a = [2,1,5,4], b = [2,1,5,4,3]

Example 3:

Input: root = [5,2,3,null,1], val = 4 Output: [5,2,4,null,1,3] Explanation: a = [2,1,5,3], b = [2,1,5,3,4]

Constraints:

The number of nodes in the tree is in the range [1, 100]. 1 <= Node.val <= 100 All the values of the tree are unique. 1 <= val <= 100

==
Problem Number: 985 URL: https://leetcode.com/problems/check-if-word-is-valid-after-substitutions Title: 1003. Check If Word Is Valid After Substitutions Problem Description: Given a string s, determine if it is valid. A string s is valid if, starting with an empty string t = "", you can transform t into s after performing the following operation any number of times:

Insert string "abc" into any position in t. More formally, t becomes tleft + "abc" + tright, where t == tleft + tright. Note that tleft and tright may be empty.

Return true if s is a valid string, otherwise, return false. Example 1: Input: s = "aabcbc" Output: true Explanation: "" -> "abc" -> "aabcbc" Thus, "aabcbc" is valid. Example 2: Input: s = "abcabcababcc" Output: true Explanation: "" -> "abc" -> "abcabc" -> "abcabcabc" -> "abcabcababcc" Thus, "abcabcababcc" is valid.

Example 3: Input: s = "abccba" Output: false Explanation: It is impossible to get "abccba" using the operation.

Constraints:

1 <= s.length <= 2 * 104 s consists of letters 'a', 'b', and 'c'

==
Problem Number: 986 URL: https://leetcode.com/problems/max-consecutive-ones-iii Title: 1004. Max Consecutive Ones III Problem Description: Given a binary array nums and an integer k, return the maximum number of consecutive 1's in the array if you can flip at most k 0's. Example 1: Input: nums = [1,1,1,0,0,0,1,1,1,1,0], k = 2 Output: 6 Explanation: [1,1,1,0,0,1,1,1,1,1,1] Bolded numbers were flipped from 0 to 1. The longest subarray is underlined. Example 2: Input: nums = [0,0,1,1,0,0,1,1,1,0,1,1,0,0,0,1,1,1,1], k = 3 Output: 10 Explanation: [0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1] Bolded numbers were flipped from 0 to 1. The longest subarray is underlined.

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1. 0 <= k <= nums.length

==
Problem Number: 987 URL: https://leetcode.com/problems/clumsy-factorial Title: 1006. Clumsy Factorial Problem Description: The factorial of a positive integer n is the product of all positive integers less than or equal to n.

For example, factorial(10) = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1.

We make a clumsy factorial using the integers in decreasing order by swapping out the multiply operations for a fixed rotation of operations with multiply '*', divide '/', add '+', and subtract '-' in this order.

For example, clumsy(10) = 10 * 9 / 8 + 7 - 6 * 5 / 4 + 3 - 2 * 1.

However, these operations are still applied using the usual order of operations of arithmetic. We do all multiplication and division steps before any addition or subtraction steps, and multiplication and division steps are processed left to right. Additionally, the division that we use is floor division such that 10 * 9 / 8 = 90 / 8 = 11. Given an integer n, return the clumsy factorial of n.   Example 1: Input: n = 4 Output: 7 Explanation: 7 = 4 * 3 / 2 + 1

Example 2: Input: n = 10 Output: 12 Explanation: 12 = 10 * 9 / 8 + 7 - 6 * 5 / 4 + 3 - 2 * 1

Constraints:

1 <= n <= 104

==
Problem Number: 988 URL: https://leetcode.com/problems/minimum-domino-rotations-for-equal-row Title: 1007. Minimum Domino Rotations For Equal Row Problem Description: In a row of dominoes, tops[i] and bottoms[i] represent the top and bottom halves of the ith domino. (A domino is a tile with two numbers from 1 to 6 - one on each half of the tile.) We may rotate the ith domino, so that tops[i] and bottoms[i] swap values. Return the minimum number of rotations so that all the values in tops are the same, or all the values in bottoms are the same. If it cannot be done, return -1.   Example 1:

Input: tops = [2,1,2,4,2,2], bottoms = [5,2,6,2,3,2] Output: 2 Explanation: The first figure represents the dominoes as given by tops and bottoms: before we do any rotations. If we rotate the second and fourth dominoes, we can make every value in the top row equal to 2, as indicated by the second figure.

Example 2: Input: tops = [3,5,1,2,3], bottoms = [3,6,3,3,4] Output: -1 Explanation: In this case, it is not possible to rotate the dominoes to make one row of values equal.

Constraints:

2 <= tops.length <= 2 * 104 bottoms.length == tops.length 1 <= tops[i], bottoms[i] <= 6

==
Problem Number: 989 URL: https://leetcode.com/problems/construct-binary-search-tree-from-preorder-traversal Title: 1008. Construct Binary Search Tree from Preorder Traversal Problem Description: Given an array of integers preorder, which represents the preorder traversal of a BST (i.e., binary search tree), construct the tree and return its root. It is guaranteed that there is always possible to find a binary search tree with the given requirements for the

given test cases. A binary search tree is a binary tree where for every node, any descendant of Node.left has a value strictly less than Node.val, and any descendant of Node.right has a value strictly greater than Node.val. A preorder traversal of a binary tree displays the value of the node first, then traverses Node.left, then traverses Node.right.   Example 1:

Input: preorder = [8,5,1,7,10,12] Output: [8,5,10,1,7,null,12]

Example 2: Input: preorder = [1,3] Output: [1,null,3]

Constraints:

1 <= preorder.length <= 100 1 <= preorder[i] <= 1000 All the values of preorder are unique.

==
Problem Number: 990 URL: https://leetcode.com/problems/pairs-of-songs-with-total-durations-divisible-by-60 Title: 1010. Pairs of Songs With Total Durations Divisible by 60 Problem Description: You are given a list of songs where the ith song has a duration of time[i] seconds. Return the number of pairs of songs for which their total duration in seconds is divisible by 60. Formally, we want the number of indices i, j such that i < j with (time[i] + time[j]) % 60 == 0.   Example 1: Input: time = [30,20,150,100,40] Output: 3 Explanation: Three pairs have a total duration divisible by 60: (time[0] = 30, time[2] = 150): total duration 180 (time[1] = 20, time[3] = 100): total duration 120 (time[1] = 20, time[4] = 40): total duration 60

Example 2: Input: time = [60,60,60] Output: 3 Explanation: All three pairs have a total duration of 120, which is divisible by 60.

Constraints:

1 <= time.length <= 6 * 104 1 <= time[i] <= 500

==
Problem Number: 991 URL: https://leetcode.com/problems/capacity-to-ship-packages-within-d-days Title: 1011. Capacity To Ship Packages Within D Days Problem Description: A conveyor belt has packages that must be shipped from one port to another within days days. The ith package on the conveyor belt has a weight of weights[i]. Each day, we load the ship with packages on the conveyor belt (in the order given by weights). We may not load more weight than the maximum weight capacity of the ship. Return the least weight capacity of the ship that will result in all the packages on the conveyor belt being shipped within days days.   Example 1: Input: weights = [1,2,3,4,5,6,7,8,9,10], days = 5 Output: 15 Explanation: A ship capacity of 15 is the minimum to ship all the packages in 5 days like this: 1st day: 1, 2, 3, 4, 5 2nd day: 6, 7 3rd day: 8 4th day: 9 5th day: 10

Note that the cargo must be shipped in the order given, so using a ship of capacity 14 and splitting the packages into parts like (2, 3, 4, 5), (1, 6, 7), (8), (9), (10) is not allowed.

Example 2: Input: weights = [3,2,2,4,1,4], days = 3 Output: 6 Explanation: A ship capacity of 6 is the minimum to ship all the packages in 3 days like this: 1st day: 3, 2 2nd day: 2, 4 3rd day: 1, 4

Example 3: Input: weights = [1,2,3,1,1], days = 4 Output: 3 Explanation: 1st day: 1 2nd day: 2 3rd day: 3 4th day: 1, 1

Constraints:

1 <= days <= weights.length <= 5 * 104 1 <= weights[i] <= 500

==

Problem Number: 992 URL: https://leetcode.com/problems/best-sightseeing-pair Title: 1014. Best Sightseeing Pair Problem Description: You are given an integer array values where values[i] represents the value of the ith sightseeing spot. Two sightseeing spots i and j have a distance j - i between them. The score of a pair (i < j) of sightseeing spots is values[i] + values[j] + i - j: the sum of the values of the sightseeing spots, minus the distance between them. Return the maximum score of a pair of sightseeing spots. Example 1: Input: values = [8,1,5,2,6] Output: 11 Explanation: i = 0, j = 2, values[i] + values[j] + i - j = 8 + 5 + 0 - 2 = 11

Example 2: Input: values = [1,2] Output: 2

Constraints:

2 <= values.length <= 5 * 104 1 <= values[i] <= 1000

==

Problem Number: 993 URL: https://leetcode.com/problems/smallest-integer-divisible-by-k Title: 1015. Smallest Integer Divisible by K Problem Description: Given a positive integer k, you need to find the length of the smallest positive integer n such that n is divisible by k, and n only contains the digit 1. Return the length of n. If there is no such n, return -1. Note: n may not fit in a 64-bit signed integer. Example 1: Input: k = 1 Output: 1 Explanation: The smallest answer is n = 1, which has length 1.

Example 2: Input: k = 2 Output: -1 Explanation: There is no such positive integer n divisible by 2.

Example 3: Input: k = 3 Output: 3 Explanation: The smallest answer is n = 111, which has length 3.

Constraints:

1 <= k <= 105

==

Problem Number: 994 URL: https://leetcode.com/problems/binary-string-with-substrings-representing-1-to-n Title: 1016. Binary String With Substrings Representing 1 To N Problem Description: Given a binary string s and a positive integer n, return true if the binary representation of all the integers in the range [1, n] are substrings of s, or false otherwise. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = ”0110”, n = 3 Output: true Example 2: Input: s = ”0110”, n = 4 Output: false

Constraints:

1 <= s.length <= 1000 s[i] is either ’0’ or ’1’. 1 <= n <= 109

==
Problem Number: 995 URL: https://leetcode.com/problems/convert-to-base-2 Title: 1017. Convert to Base -2 Problem Description: Given an integer n, return a binary string representing its representation in base -2. Note that the returned string should not have leading zeros unless the string is ”0”. Example 1: Input: n = 2 Output: ”110” Explantion: (-2)2 + (-2)1 = 2

Example 2: Input: n = 3 Output: ”111” Explantion: (-2)2 + (-2)1 + (-2)0 = 3

Example 3: Input: n = 4 Output: ”100” Explantion: (-2)2 = 4

Constraints:

0 <= n <= 109

==
Problem Number: 996 URL: https://leetcode.com/problems/next-greater-node-in-linked-list Title: 1019. Next Greater Node In Linked List Problem Description: You are given the head of a linked list with n nodes. For each node in the list, find the value of the next greater node. That is, for each node, find the value of the first node that is next to it and has a strictly larger value than it. Return an integer array answer where answer[i] is the value of the next greater node of the ith node (1-indexed). If the ith node does not have a next greater node, set answer[i] = 0. Example 1:

Input: head = [2,1,5] Output: [5,5,0]

Example 2:

Input: head = [2,7,4,3,5] Output: [7,0,5,5,0]

Constraints:

The number of nodes in the list is n. 1 <= n <= 104 1 <= Node.val <= 109

==
Problem Number: 997 URL: https://leetcode.com/problems/number-of-enclaves Title: 1020. Number of Enclaves Problem Description: You are given an m x n binary matrix grid, where 0 represents a sea cell and 1 represents a land cell. A move consists of walking from one land cell to another adjacent (4-directionally) land cell or walking off the boundary of the grid. Return the number of land cells in grid for which we cannot walk off the boundary of the grid in any number of moves. Example 1:

Input: grid = [[0,0,0,0],[1,0,1,0],[0,1,1,0],[0,0,0,0]] Output: 3 Explanation: There are three 1s that are enclosed by 0s, and one 1 that is not enclosed because its on the boundary.

Example 2:

Input: grid = [[0,1,1,0],[0,0,1,0],[0,0,1,0],[0,0,0,0]] Output: 0 Explanation: All 1s are either on the boundary or can reach the boundary.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 500 grid[i][j] is either 0 or 1.

==
Problem Number: 998 URL: https://leetcode.com/problems/camelcase-matching Title: 1023. Camelcase Matching Problem Description: Given an array of strings queries and a string pattern, return a boolean array answer where answer[i] is true if queries[i] matches pattern, and false otherwise. A query word queries[i] matches pattern if you can insert lowercase English letters pattern so that it equals the query. You may insert each character at any position and you may not insert any characters. Example 1: Input: queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FB" Output: [true,false,true,true,false] Explanation: "FooBar" can be generated like this "F" + "oo" + "B" + "ar". "FootBall" can be generated like this "F" + "oot" + "B" + "all". "FrameBuffer" can be generated like this "F" + "rame" + "B" + "uffer".

Example 2: Input: queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FoBa" Output: [true,false,true,false,false] Explanation: "FooBar" can be generated like this "Fo" + "o" + "Ba" + "r". "FootBall" can be generated like this "Fo" + "ot" + "Ba" + "ll".

Example 3: Input: queries = ["FooBar","FooBarTest","FootBall","FrameBuffer","ForceFeedBack"], pattern = "FoBaT" Output: [false,true,false,false,false] Explanation: "FooBarTest" can be generated like this "Fo" + "o" + "Ba" + "r" + "T" + "est".

Constraints:

1 <= pattern.length, queries.length <= 100 1 <= queries[i].length <= 100 queries[i] and pattern consist of English letters.

==
Problem Number: 999 URL: https://leetcode.com/problems/video-stitching Title: 1024. Video Stitching Problem Description: You are given a series of video clips from a sporting event that lasted time seconds. These video clips can be overlapping with each other and have varying lengths. Each video clip is described by an array clips where clips[i] = [starti, endi] indicates that the ith clip started at starti and ended at endi. We can cut these clips into segments freely.

For example, a clip [0, 7] can be cut into segments [0, 1] + [1, 3] + [3, 7].

Return the minimum number of clips needed so that we can cut the clips into segments that cover the entire sporting event [0, time]. If the task is impossible, return -1. Example 1: Input: clips = [[0,2],[4,6],[8,10],[1,9],[1,5],[5,9]], time = 10 Output: 3 Explanation: We take the clips [0,2], [8,10], [1,9]; a total of 3 clips. Then, we can reconstruct the sporting event as follows: We cut [1,9] into segments [1,2] + [2,8] + [8,9]. Now we have segments [0,2] + [2,8] + [8,10] which cover the sporting event [0, 10].

Example 2: Input: clips = [[0,1],[1,2]], time = 5 Output: -1 Explanation: We cannot cover [0,5] with only [0,1] and [1,2].

Example 3: Input: clips = [[0,1],[6,8],[0,2],[5,6],[0,4],[0,3],[6,7],[1,3],[4,7],[1,4],[2,5],[2,6],[3,4],[4,5],[5,7],[6,9]], time = 9 Output: 3 Explanation: We can take clips [0,4], [4,7], and [6,9].

Constraints:

1 <= clips.length <= 100 0 <= starti <= endi <= 100 1 <= time <= 100

==

Problem Number: 1000 URL: https://leetcode.com/problems/maximum-difference-between-node-and-ancestor Title: 1026. Maximum Difference Between Node and Ancestor Problem Description: Given the root of a binary tree, find the maximum value v for which there exist different nodes a and b where v = |a.val - b.val| and a is an ancestor of b. A node a is an ancestor of b if either: any child of a is equal to b or any child of a is an ancestor of b. Example 1:

Input: root = [8,3,10,1,6,null,14,null,null,4,7,13] Output: 7 Explanation: We have various ancestor-node differences, some of which are given below : |8 - 3| = 5 |3 - 7| = 4 |8 - 1| = 7 |10 - 13| = 3 Among all possible differences, the maximum value of 7 is obtained by |8 - 1| = 7. Example 2:

Input: root = [1,null,2,null,0,3] Output: 3

Constraints:

The number of nodes in the tree is in the range [2, 5000]. 0 <= Node.val <= 105

==

Problem Number: 1001 URL: https://leetcode.com/problems/longest-arithmetic-subsequence Title: 1027. Longest Arithmetic Subsequence Problem Description: Given an array nums of integers, return the length of the longest arithmetic subsequence in nums. Note that:

A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements. A sequence seq is arithmetic if seq[i + 1] - seq[i] are all the same value (for 0 <= i < seq.length - 1).

Example 1: Input: nums = [3,6,9,12] Output: 4 Explanation: The whole array is an arithmetic sequence with steps of length = 3.

Example 2: Input: nums = [9,4,7,2,10] Output: 3 Explanation: The longest arithmetic subsequence is [4,7,10].

Example 3: Input: nums = [20,1,15,3,10,5,8] Output: 4 Explanation: The longest arithmetic subsequence is [20,15,10,5].

Constraints:

2 <= nums.length <= 1000 0 <= nums[i] <= 500

==
Problem Number: 1002 URL: https://leetcode.com/problems/two-city-scheduling Title: 1029. Two City Scheduling Problem Description: A company is planning to interview 2n people. Given the array costs where costs[i] = [aCosti, bCosti], the cost of flying the ith person to city a is aCosti, and the cost of flying the ith person to city b is bCosti. Return the minimum cost to fly every person to a city such that exactly n people arrive in each city. Example 1: Input: costs = [[10,20],[30,200],[400,50],[30,20]] Output: 110 Explanation: The first person goes to city A for a cost of 10. The second person goes to city A for a cost of 30. The third person goes to city B for a cost of 50. The fourth person goes to city B for a cost of 20.

The total minimum cost is 10 + 30 + 50 + 20 = 110 to have half the people interviewing in each city.

Example 2: Input: costs = [[259,770],[448,54],[926,667],[184,139],[840,118],[577,469]] Output: 1859

Example 3: Input: costs = [[515,563],[451,713],[537,709],[343,819],[855,779],[457,60],[650,359],[631,42]] Output: 3086

Constraints:

2 * n == costs.length 2 <= costs.length <= 100 costs.length is even. 1 <= aCosti, bCosti <= 1000

==
Problem Number: 1003 URL: https://leetcode.com/problems/maximum-sum-of-two-non-overlapping-subarrays Title: 1031. Maximum Sum of Two Non-Overlapping Subarrays Problem Description: Given an integer array nums and two integers firstLen and secondLen, return the maximum sum of elements in two non-overlapping subarrays with lengths firstLen and secondLen. The array with length firstLen could occur before or after the array with length secondLen, but they have to be non-overlapping. A subarray is a contiguous part of an array. Example 1: Input: nums = [0,6,5,2,2,5,1,9,4], firstLen = 1, secondLen = 2 Output: 20 Explanation: One choice of subarrays is [9] with length 1, and [6,5] with length 2.

Example 2: Input: nums = [3,8,1,3,2,1,8,9,0], firstLen = 3, secondLen = 2 Output: 29 Explanation: One choice of subarrays is [3,8,1] with length 3, and [8,9] with length 2.

Example 3: Input: nums = [2,1,5,6,0,9,5,0,3,8], firstLen = 4, secondLen = 3 Output: 31 Explanation: One choice of subarrays is [5,6,0,9] with length 4, and [0,3,8] with length 3.

Constraints:

1 <= firstLen, secondLen <= 1000 2 <= firstLen + secondLen <= 1000 firstLen + secondLen <= nums.length <= 1000 0 <= nums[i] <= 1000

==
Problem Number: 1004 URL: https://leetcode.com/problems/moving-stones-until-consecutive Title: 1033. Moving Stones Until Consecutive Problem Description: There are three stones in different positions on the X-axis. You are given three integers a, b, and c, the positions of the stones. In one move, you pick up a stone at an endpoint (i.e., either the lowest or highest position stone), and move it to an unoccupied position between those endpoints. Formally, let's say the stones are currently at positions x, y, and z with x < y < z. You pick up the stone at either position x or position z, and move that stone to an integer position k, with x < k < z and k != y. The game ends when you cannot make any more moves (i.e., the stones are in three consecutive positions). Return an integer array answer of length 2 where:

answer[0] is the minimum number of moves you can play, and answer[1] is the maximum number of moves you can play.

Example 1: Input: a = 1, b = 2, c = 5 Output: [1,2] Explanation: Move the stone from 5 to 3, or move the stone from 5 to 4 to 3.

Example 2: Input: a = 4, b = 3, c = 2 Output: [0,0] Explanation: We cannot make any moves.

Example 3: Input: a = 3, b = 5, c = 1 Output: [1,2] Explanation: Move the stone from 1 to 4; or move the stone from 1 to 2 to 4.

Constraints:

1 <= a, b, c <= 100 a, b, and c have different values.

==
Problem Number: 1005 URL: https://leetcode.com/problems/coloring-a-border Title: 1034. Coloring A Border Problem Description: You are given an m x n integer matrix grid, and three integers row, col, and color. Each value in the grid represents the color of the grid square at that location. Two squares are called adjacent if they are next to each other in any of the 4 directions. Two squares belong to the same connected component if they have the same color and they are adjacent. The border of a connected component is all the squares in the connected component that are either adjacent to (at least) a

square not in the component, or on the boundary of the grid (the first or last row or column). You should color the border of the connected component that contains the square grid[row][col] with color. Return the final grid. Example 1: Input: grid = [[1,1],[1,2]], row = 0, col = 0, color = 3 Output: [[3,3],[3,2]] Example 2: Input: grid = [[1,2,2],[2,3,2]], row = 0, col = 1, color = 3 Output: [[1,3,3],[2,3,3]] Example 3: Input: grid = [[1,1,1],[1,1,1],[1,1,1]], row = 1, col = 1, color = 2 Output: [[2,2,2],[2,1,2],[2,2,2]]

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 50 1 <= grid[i][j], color <= 1000 0 <= row < m 0 <= col < n

==
Problem Number: 1006 URL: https://leetcode.com/problems/uncrossed-lines Title: 1035. Uncrossed Lines Problem Description: You are given two integer arrays nums1 and nums2. We write the integers of nums1 and nums2 (in the order they are given) on two separate horizontal lines. We may draw connecting lines: a straight line connecting two numbers nums1[i] and nums2[j] such that:

nums1[i] == nums2[j], and the line we draw does not intersect any other connecting (non-horizontal) line.

Note that a connecting line cannot intersect even at the endpoints (i.e., each number can only belong to one connecting line). Return the maximum number of connecting lines we can draw in this way. Example 1:

Input: nums1 = [1,4,2], nums2 = [1,2,4] Output: 2 Explanation: We can draw 2 uncrossed lines as in the diagram. We cannot draw 3 uncrossed lines, because the line from nums1[1] = 4 to nums2[2] = 4 will intersect the line from nums1[2]=2 to nums2[1]=2.

Example 2: Input: nums1 = [2,5,1,2,5], nums2 = [10,5,2,1,5,2] Output: 3

Example 3: Input: nums1 = [1,3,7,1,7,5], nums2 = [1,9,2,5,1] Output: 2

Constraints:

1 <= nums1.length, nums2.length <= 500 1 <= nums1[i], nums2[j] <= 2000

==
Problem Number: 1007 URL: https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree Title: 1038. Binary Search Tree to Greater Sum Tree Problem Description: Given the root of a Binary Search Tree (BST), convert it to a Greater Tree such that every key of the original BST is changed to the original key plus the sum of all keys greater than the original key in BST. As a reminder, a binary search tree is a tree that satisfies these constraints:

The left subtree of a node contains only nodes with keys less than the node's key. The right subtree of a node contains only nodes with keys greater than the node's key. Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [4,1,6,0,2,5,7,null,null,null,3,null,null,null,8] Output: [30,36,21,36,35,26,15,null,null,null,33,null,nu

Example 2: Input: root = [0,null,1] Output: [1,null,1]

Constraints:

The number of nodes in the tree is in the range [1, 100]. 0 <= Node.val <= 100 All the values in the tree are unique.

Note: This question is the same as 538: https://leetcode.com/problems/convert-bst-to-greater-tree/

==
Problem Number: 1008 URL: https://leetcode.com/problems/minimum-score-triangulation-of-polygon Title: 1039. Minimum Score Triangulation of Polygon Problem Description: You have a convex n-sided polygon where each vertex has an integer value. You are given an integer array values where values[i] is the value of the ith vertex (i.e., clockwise order). You will triangulate the polygon into n - 2 triangles. For each triangle, the value of that triangle is the product of the values of its vertices, and the total score of the triangulation is the sum of these values over all n - 2 triangles in the triangulation. Return the smallest possible total score that you can achieve with some triangulation of the polygon. Example 1:

Input: values = [1,2,3] Output: 6 Explanation: The polygon is already triangulated, and the score of the only triangle is 6.

Example 2:

Input: values = [3,7,4,5] Output: 144 Explanation: There are two triangulations, with possible scores: 3*7*5 + 4*5*7 = 245, or 3*4*5 + 3*4*7 = 144. The minimum score is 144.

Example 3:

Input: values = [1,3,1,4,1,5] Output: 13 Explanation: The minimum score triangulation has score 1*1*3 + 1*1*4 + 1*1*5 + 1*1*1 = 13.

Constraints:

n == values.length 3 <= n <= 50 1 <= values[i] <= 100

==
Problem Number: 1009 URL: https://leetcode.com/problems/moving-stones-until-consecutive-ii Title: 1040. Moving Stones Until Consecutive II Problem Description: There are some stones in different positions on the X-axis. You are given an integer array stones, the positions of the stones. Call a stone an endpoint stone if it has the smallest or largest position. In one move, you pick up an endpoint stone and move it to an unoccupied position so that it is no longer an endpoint stone.

In particular, if the stones are at say, stones = [1,2,5], you cannot move the endpoint stone at position 5, since moving it to any position (such as 0, or 3) will still keep that stone as an endpoint stone.

The game ends when you cannot make any more moves (i.e., the stones are in three consecutive positions). Return an integer array answer of length 2 where:

answer[0] is the minimum number of moves you can play, and answer[1] is the maximum number of moves you can play.

Example 1: Input: stones = [7,4,9] Output: [1,2] Explanation: We can move 4 -> 8 for one move to finish the game. Or, we can move 9 -> 5, 4 -> 6 for two moves to finish the game.

Example 2: Input: stones = [6,5,4,3,10] Output: [2,3] Explanation: We can move 3 -> 8 then 10 -> 7 to finish the game. Or, we can move 3 -> 7, 4 -> 8, 5 -> 9 to finish the game. Notice we cannot move 10 -> 2 to finish the game, because that would be an illegal move.

Constraints:

3 <= stones.length <= 104 1 <= stones[i] <= 109 All the values of stones are unique.

==
Problem Number: 1010 URL: https://leetcode.com/problems/robot-bounded-in-circle Title: 1041. Robot Bounded In Circle Problem Description: On an infinite plane, a robot initially stands at (0, 0) and faces north. Note that:

The north direction is the positive direction of the y-axis. The south direction is the negative direction of the y-axis. The east direction is the positive direction of the x-axis. The west direction is the negative direction of the x-axis.

The robot can receive one of three instructions:

"G": go straight 1 unit. "L": turn 90 degrees to the left (i.e., anti-clockwise direction). "R": turn 90 degrees to the right (i.e., clockwise direction).

The robot performs the instructions given in order, and repeats them forever. Return true if and only if there exists a circle in the plane such that the robot never leaves the circle. Example 1: Input: instructions = "GGLLGG" Output: true Explanation: The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "G": move one step. Position: (0, 2). Direction: North. "L": turn 90 degrees anti-clockwise. Position: (0, 2). Direction: West. "L": turn 90 degrees anti-clockwise. Position: (0, 2). Direction: South. "G": move one step. Position: (0, 1). Direction: South. "G": move one step. Position: (0, 0). Direction: South. Repeating the instructions, the robot goes into the cycle: (0, 0) --> (0, 1) --> (0, 2) --> (0, 1) --> (0, 0). Based on that, we return true.

Example 2: Input: instructions = "GG" Output: false Explanation: The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "G": move one step. Position: (0, 2). Direction: North. Repeating the instructions, keeps advancing in the north direction and does not go into cycles. Based on that, we return false.

Example 3: Input: instructions = "GL" Output: true Explanation: The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "L": turn 90 degrees anti-clockwise. Position: (0, 1). Direction: West. "G": move one step. Position: (-1, 1). Direction: West. "L": turn 90 degrees anti-clockwise. Position: (-1, 1). Direction: South. "G": move one step. Position: (-1, 0). Direction: South. "L": turn 90 degrees anti-clockwise. Position: (-1, 0). Direction: East. "G": move one step. Position: (0, 0). Direction: East. "L": turn 90 degrees anti-clockwise. Position: (0, 0). Direction: North. Repeating the instructions, the robot goes into the cycle: (0, 0) --> (0, 1) --> (-1, 1) --> (-1, 0) --> (0, 0). Based on that, we return true.

Constraints:

1 <= instructions.length <= 100 instructions[i] is 'G', 'L' or, 'R'.

==
Problem Number: 1011 URL: https://leetcode.com/problems/flower-planting-with-no-adjacent Title: 1042. Flower Planting With No Adjacent Problem Description: You have n gardens, labeled from 1 to n, and an array paths where paths[i] = [xi, yi] describes a bidirectional path between garden xi to garden yi. In each garden, you want to plant one of 4 types of flowers. All gardens have at most 3 paths coming into or leaving it. Your task is to choose a flower type for each garden such that, for any two gardens connected by a path, they have different types of flowers. Return any such a choice as an array answer, where answer[i] is the type of flower planted in the (i+1)th garden. The flower types are denoted 1, 2, 3, or 4. It is guaranteed an answer exists. Example 1: Input: n = 3, paths = [[1,2],[2,3],[3,1]] Output: [1,2,3] Explanation: Gardens 1 and 2 have different types. Gardens 2 and 3 have different types. Gardens 3 and 1 have different types. Hence, [1,2,3] is a valid answer. Other valid answers include [1,2,4], [1,4,2], and [3,2,1].

Example 2: Input: n = 4, paths = [[1,2],[3,4]] Output: [1,2,1,2]

Example 3: Input: n = 4, paths = [[1,2],[2,3],[3,4],[4,1],[1,3],[2,4]] Output: [1,2,3,4]

Constraints:

1 <= n <= 104 0 <= paths.length <= 2 * 104 paths[i].length == 2 1 <= xi, yi <= n xi != yi Every garden has at most 3 paths coming into or leaving it.

==
Problem Number: 1012 URL: https://leetcode.com/problems/partition-array-for-maximum-sum Title: 1043. Partition Array for Maximum Sum Problem Description: Given an integer array arr, partition the array into (contiguous) subarrays of length at most k. After partitioning, each subarray has their values

changed to become the maximum value of that subarray. Return the largest sum of the given array after partitioning. Test cases are generated so that the answer fits in a 32-bit integer.   Example 1: Input: arr = [1,15,7,9,2,5,10], k = 3 Output: 84 Explanation: arr becomes [15,15,15,9,10,10,10]

Example 2: Input: arr = [1,4,1,5,7,3,6,1,9,9,3], k = 4 Output: 83

Example 3: Input: arr = [1], k = 1 Output: 1

Constraints:

1 <= arr.length <= 500 0 <= arr[i] <= 109 1 <= k <= arr.length

==
Problem Number: 1013 URL: https://leetcode.com/problems/longest-string-chain Title: 1048. Longest String Chain Problem Description: You are given an array of words where each word consists of lowercase English letters. wordA is a predecessor of wordB if and only if we can insert exactly one letter anywhere in wordA without changing the order of the other characters to make it equal to wordB.

For example, "abc" is a predecessor of "abac", while "cba" is not a predecessor of "bcad".

A word chain is a sequence of words [word1, word2, ..., wordk] with k >= 1, where word1 is a predecessor of word2, word2 is a predecessor of word3, and so on. A single word is trivially a word chain with k == 1. Return the length of the longest possible word chain with words chosen from the given list of words.   Example 1: Input: words = ["a","b","ba","bca","bda","bdca"] Output: 4 Explanation: One of the longest word chains is ["a","ba","bda","bdca"].

Example 2: Input: words = ["xbc","pcxbcf","xb","cxbc","pcxbc"] Output: 5 Explanation: All the words can be put in a word chain ["xb", "xbc", "cxbc", "pcxbc", "pcxbcf"].

Example 3: Input: words = ["abcd","dbqca"] Output: 1 Explanation: The trivial word chain ["abcd"] is one of the longest word chains. ["abcd","dbqca"] is not a valid word chain because the ordering of the letters is changed.

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 16 words[i] only consists of lowercase English letters.

==
Problem Number: 1014 URL: https://leetcode.com/problems/last-stone-weight-ii Title: 1049. Last Stone Weight II Problem Description: You are given an array of integers stones where stones[i] is the weight of the ith stone. We are playing a game with the stones. On each turn, we choose any two stones and smash them together. Suppose the stones have weights x and y with x <= y. The result of this smash is:

If x == y, both stones are destroyed, and If x != y, the stone of weight x is destroyed, and the stone of weight y has new weight y - x.

At the end of the game, there is at most one stone left. Return the smallest possible weight of the left stone. If there are no stones left, return 0. Example 1: Input: stones = [2,7,4,1,8,1] Output: 1 Explanation: We can combine 2 and 4 to get 2, so the array converts to [2,7,1,8,1] then, we can combine 7 and 8 to get 1, so the array converts to [2,1,1,1] then, we can combine 2 and 1 to get 1, so the array converts to [1,1,1] then, we can combine 1 and 1 to get 0, so the array converts to [1], then that's the optimal value.

Example 2: Input: stones = [31,26,33,21,40] Output: 5

Constraints:

1 <= stones.length <= 30 1 <= stones[i] <= 100

== Problem Number: 1015 URL: https://leetcode.com/problems/grumpy-bookstore-owner Title: 1052. Grumpy Bookstore Owner Problem Description: There is a bookstore owner that has a store open for n minutes. Every minute, some number of customers enter the store. You are given an integer array customers of length n where customers[i] is the number of the customer that enters the store at the start of the ith minute and all those customers leave after the end of that minute. On some minutes, the bookstore owner is grumpy. You are given a binary array grumpy where grumpy[i] is 1 if the bookstore owner is grumpy during the ith minute, and is 0 otherwise. When the bookstore owner is grumpy, the customers of that minute are not satisfied, otherwise, they are satisfied. The bookstore owner knows a secret technique to keep themselves not grumpy for minutes consecutive minutes, but can only use it once. Return the maximum number of customers that can be satisfied throughout the day. Example 1: Input: customers = [1,0,1,2,1,1,7,5], grumpy = [0,1,0,1,0,1,0,1], minutes = 3 Output: 16 Explanation: The bookstore owner keeps themselves not grumpy for the last 3 minutes. The maximum number of customers that can be satisfied = 1 + 1 + 1 + 1 + 7 + 5 = 16.

Example 2: Input: customers = [1], grumpy = [0], minutes = 1 Output: 1

Constraints:

n == customers.length == grumpy.length 1 <= minutes <= n <= 2 * 104 0 <= customers[i] <= 1000 grumpy[i] is either 0 or 1.

== Problem Number: 1016 URL: https://leetcode.com/problems/previous-permutation-with-one-swap Title: 1053. Previous Permutation With One Swap Problem Description: Given an array of positive integers arr (not necessarily distinct), return the lexicographically largest permutation that is smaller than arr, that can be made with exactly one swap. If it cannot be done, then return the same array. Note that a swap exchanges the positions of two numbers arr[i] and arr[j]   Example 1: Input: arr = [3,2,1] Output: [3,1,2] Explanation: Swapping 2 and 1.

Example 2: Input: arr = [1,1,5] Output: [1,1,5] Explanation: This is already the smallest permutation.

Example 3: Input: arr = [1,9,4,6,7] Output: [1,7,4,6,9] Explanation: Swapping 9 and 7.

Constraints:

1 <= arr.length <= 104 1 <= arr[i] <= 104

==
Problem Number: 1017 URL: https://leetcode.com/problems/distant-barcodes Title: 1054. Distant Barcodes Problem Description: In a warehouse, there is a row of barcodes, where the ith barcode is barcodes[i]. Rearrange the barcodes so that no two adjacent barcodes are equal. You may return any answer, and it is guaranteed an answer exists.   Example 1: Input: barcodes = [1,1,1,2,2,2] Output: [2,1,2,1,2,1] Example 2: Input: barcodes = [1,1,1,1,2,2,3,3] Output: [1,3,1,3,1,2,1,2]

Constraints:

1 <= barcodes.length <= 10000 1 <= barcodes[i] <= 10000

==
Problem Number: 1018 URL: https://leetcode.com/problems/lexicographically-smallest-equivalent-string Title: 1061. Lexicographically Smallest Equivalent String Problem Description: You are given two strings of the same length s1 and s2 and a string baseStr. We say s1[i] and s2[i] are equivalent characters.

For example, if s1 = "abc" and s2 = "cde", then we have 'a' == 'c', 'b' == 'd', and 'c' == 'e'.

Equivalent characters follow the usual rules of any equivalence relation:

Reflexivity: 'a' == 'a'. Symmetry: 'a' == 'b' implies 'b' == 'a'. Transitivity: 'a' == 'b' and 'b' == 'c' implies 'a' == 'c'.

For example, given the equivalency information from s1 = "abc" and s2 = "cde", "acd" and "aab" are equivalent strings of baseStr = "eed", and "aab" is the lexicographically smallest equivalent string of baseStr. Return the lexicographically smallest equivalent string of baseStr by using the equivalency information from s1 and s2.   Example 1: Input: s1 = "parker", s2 = "morris", baseStr = "parser" Output: "makkek" Explanation: Based on the equivalency information in s1 and s2, we can group their characters as [m,p], [a,o], [k,r,s], [e,i]. The characters in each group are equivalent and sorted in lexicographical order. So the answer is "makkek".

Example 2: Input: s1 = "hello", s2 = "world", baseStr = "hold" Output: "hdld" Explanation: Based on the equivalency information in s1 and s2, we can group their characters as [h,w], [d,e,o], [l,r]. So only the second letter 'o' in baseStr is changed to 'd', the answer is "hdld".

Example 3: Input: s1 = "leetcode", s2 = "programs", baseStr = "sourcecode" Output: "aauaaaaada" Explanation: We group the equivalent characters in s1 and s2 as [a,o,e,r,s,c], [l,p], [g,t] and [d,m], thus all letters in baseStr except 'u' and 'd' are transformed to 'a', the answer is "aauaaaaada".

Constraints:

1 <= s1.length, s2.length, baseStr <= 1000 s1.length == s2.length s1, s2, and baseStr consist of lowercase English letters.

==
Problem Number: 1019 URL: https://leetcode.com/problems/flip-columns-for-maximum-number-of-equal-rows Title: 1072. Flip Columns For Maximum Number of Equal Rows Problem Description: You are given an m x n binary matrix matrix. You can choose any number of columns in the matrix and flip every cell in that column (i.e., Change the value of the cell from 0 to 1 or vice versa). Return the maximum number of rows that have all values equal after some number of flips. Example 1: Input: matrix = [[0,1],[1,1]] Output: 1 Explanation: After flipping no values, 1 row has all values equal.

Example 2: Input: matrix = [[0,1],[1,0]] Output: 2 Explanation: After flipping values in the first column, both rows have equal values.

Example 3: Input: matrix = [[0,0,0],[0,0,1],[1,1,0]] Output: 2 Explanation: After flipping values in the first two columns, the last two rows have equal values.

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 300 matrix[i][j] is either 0 or 1.

==
Problem Number: 1020 URL: https://leetcode.com/problems/adding-two-negabinary-numbers Title: 1073. Adding Two Negabinary Numbers Problem Description: Given two numbers arr1 and arr2 in base -2, return the result of adding them together. Each number is given in array format: as an array of 0s and 1s, from most significant bit to least significant bit. For example, arr = [1,1,0,1] represents the number (-2)^3 + (-2)^2 + (-2)^0 = -3. A number arr in array, format is also guaranteed to have no leading zeros: either arr == [0] or arr[0] == 1. Return the result of adding arr1 and arr2 in the same format: as an array of 0s and 1s with no leading zeros. Example 1: Input: arr1 = [1,1,1,1,1], arr2 = [1,0,1] Output: [1,0,0,0,0] Explanation: arr1 represents 11, arr2 represents 5, the output represents 16.

Example 2: Input: arr1 = [0], arr2 = [0] Output: [0]

Example 3: Input: arr1 = [0], arr2 = [1] Output: [1]

Constraints:

1 <= arr1.length, arr2.length <= 1000 arr1[i] and arr2[i] are 0 or 1 arr1 and arr2 have no leading zeros

==
Problem Number: 1021 URL: https://leetcode.com/problems/letter-tile-possibilities Title: 1079. Letter Tile Possibilities Problem Description: You have n tiles, where each tile has one letter tiles[i] printed on it. Return the number of possible non-empty sequences of letters you can make using the letters printed on those tiles. Example 1: Input: tiles = "AAB" Output: 8 Explanation: The possible sequences are "A", "B", "AA", "AB", "BA", "AAB", "ABA", "BAA".

Example 2: Input: tiles = "AAABBC" Output: 188

Example 3: Input: tiles = "V" Output: 1

Constraints:

1 <= tiles.length <= 7 tiles consists of uppercase English letters.

==
Problem Number: 1022 URL: https://leetcode.com/problems/insufficient-nodes-in-root-to-leaf-paths Title: 1080. Insufficient Nodes in Root to Leaf Paths Problem Description: Given the root of a binary tree and an integer limit, delete all insufficient nodes in the tree simultaneously, and return the root of the resulting binary tree. A node is insufficient if every root to leaf path intersecting this node has a sum strictly less than limit. A leaf is a node with no children. Example 1:

Input: root = [1,2,3,4,-99,-99,7,8,9,-99,-99,12,13,-99,14], limit = 1 Output: [1,2,3,4,null,null,7,8,9,null,14]

Example 2:

Input: root = [5,4,8,11,null,17,4,7,1,null,null,5,3], limit = 22 Output: [5,4,8,11,null,17,4,7,null,null,null,5]

Example 3:

Input: root = [1,2,-3,-5,null,4,null], limit = -1 Output: [1,null,-3,4]

Constraints:

The number of nodes in the tree is in the range [1, 5000]. -105 <= Node.val <= 105 -109 <= limit <= 109

==
Problem Number: 1023 URL: https://leetcode.com/problems/smallest-subsequence-of-distinct-characters Title: 1081. Smallest Subsequence of Distinct Characters Problem Description: Given a string s, return the lexicographically smallest subsequence of s that contains all the distinct characters of s exactly once. Example 1: Input: s = "bcabc" Output: "abc"

Example 2: Input: s = ”cbacdcbc” Output: ”acdb”

Constraints:

1 <= s.length <= 1000 s consists of lowercase English letters.

Note: This question is the same as 316: https://leetcode.com/problems/remove-duplicate-letters/ == Problem Number: 1024 URL: https://leetcode.com/problems/largest-values-from-labels Title: 1090. Largest Values From Labels Problem Description: There is a set of n items. You are given two integer arrays values and labels where the value and the label of the ith element are values[i] and labels[i] respectively. You are also given two integers numWanted and useLimit. Choose a subset s of the n elements such that:

The size of the subset s is less than or equal to numWanted. There are at most useLimit items with the same label in s.

The score of a subset is the sum of the values in the subset. Return the maximum score of a subset s. Example 1: Input: values = [5,4,3,2,1], labels = [1,1,2,2,3], numWanted = 3, useLimit = 1 Output: 9 Explanation: The subset chosen is the first, third, and fifth items.

Example 2: Input: values = [5,4,3,2,1], labels = [1,3,3,3,2], numWanted = 3, useLimit = 2 Output: 12 Explanation: The subset chosen is the first, second, and third items.

Example 3: Input: values = [9,8,8,7,6], labels = [0,0,0,1,1], numWanted = 3, useLimit = 1 Output: 16 Explanation: The subset chosen is the first and fourth items.

Constraints:

n == values.length == labels.length 1 <= n <= 2 * 104 0 <= values[i], labels[i] <= 2 * 104 1 <= numWanted, useLimit <= n

== Problem Number: 1025 URL: https://leetcode.com/problems/shortest-path-in-binary-matrix Title: 1091. Shortest Path in Binary Matrix Problem Description: Given an n x n binary matrix grid, return the length of the shortest clear path in the matrix. If there is no clear path, return -1. A clear path in a binary matrix is a path from the top-left cell (i.e., (0, 0)) to the bottom-right cell (i.e., (n - 1, n - 1)) such that:

All the visited cells of the path are 0. All the adjacent cells of the path are 8-directionally connected (i.e., they are different and they share an edge or a corner).

The length of a clear path is the number of visited cells of this path. Example 1:

Input: grid = [[0,1],[1,0]] Output: 2

Example 2:

Input: grid = [[0,0,0],[1,1,0],[1,1,0]] Output: 4

Example 3: Input: grid = [[1,0,0],[1,1,0],[1,1,0]] Output: -1

Constraints:

n == grid.length n == grid[i].length 1 <= n <= 100 grid[i][j] is 0 or 1

==
Problem Number: 1026 URL: https://leetcode.com/problems/statistics-from-a-large-sample Title: 1093. Statistics from a Large Sample Problem Description: You are given a large sample of integers in the range [0, 255]. Since the sample is so large, it is represented by an array count where count[k] is the number of times that k appears in the sample. Calculate the following statistics:

minimum: The minimum element in the sample. maximum: The maximum element in the sample. mean: The average of the sample, calculated as the total sum of all elements divided by the total number of elements. median:

If the sample has an odd number of elements, then the median is the middle element once the sample is sorted. If the sample has an even number of elements, then the median is the average of the two middle elements once the sample is sorted.

mode: The number that appears the most in the sample. It is guaranteed to be unique.

Return the statistics of the sample as an array of floating-point numbers [minimum, maximum, mean, median, mode]. Answers within 10-5 of the actual answer will be accepted. Example 1: Input: count = [0,1,3,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 Output: [1.00000,3.00000,2.37500,2.50000,3.00000] Explanation: The sample represented by count is [1,2,2,2,3,3,3,3]. The minimum and maximum are 1 and 3 respectively. The mean is (1+2+2+2+3+3+3+3) / 8 = 19 / 8 = 2.375. Since the size of the sample is even, the median is the average of the two middle elements 2 and 3, which is 2.5. The mode is 3 as it appears the most in the sample.

Example 2: Input: count = [0,4,3,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, Output: [1.00000,4.00000,2.18182,2.00000,1.00000] Explanation: The sample represented by count is [1,1,1,1,2,2,2,3,3,4,4]. The minimum and maximum are 1 and 4 respectively. The mean is (1+1+1+1+2+2+2+3+3+4+4) / 11 = 24 / 11 = 2.18181818... (for display purposes, the output shows the rounded number 2.18182). Since the size of the sample is odd, the median is the middle element 2. The mode is 1 as it appears the most in the sample.

Constraints:

count.length == 256 0 <= count[i] <= 109 1 <= sum(count) <= 109 The mode of the sample that count represents is unique.

==
Problem Number: 1027 URL: https://leetcode.com/problems/car-pooling Title: 1094. Car Pooling Problem Description: There is a car with capacity empty seats. The vehicle only drives east (i.e., it cannot turn around and drive west). You are given the integer capacity and an array trips where trips[i] = [numPassengersi, fromi, toi] indicates that the ith trip has numPassengersi passengers and the locations to pick them up and drop them off are fromi and toi respectively. The locations are given as the number of kilometers due east from the car's initial location. Return true if it is possible to pick up and drop off all passengers for all the given trips, or false otherwise. Example 1: Input: trips = [[2,1,5],[3,3,7]], capacity = 4 Output: false

Example 2: Input: trips = [[2,1,5],[3,3,7]], capacity = 5 Output: true

Constraints:

1 <= trips.length <= 1000 trips[i].length == 3 1 <= numPassengersi <= 100 0 <= fromi < toi <= 1000 1 <= capacity <= 105

==
Problem Number: 1028 URL: https://leetcode.com/problems/path-in-zigzag-labelled-binary-tree Title: 1104. Path In Zigzag Labelled Binary Tree Problem Description: In an infinite binary tree where every node has two children, the nodes are labelled in row order. In the odd numbered rows (ie., the first, third, fifth,...), the labelling is left to right, while in the even numbered rows (second, fourth, sixth,...), the labelling is right to left.

Given the label of a node in this tree, return the labels in the path from the root of the tree to the node with that label. Example 1: Input: label = 14 Output: [1,3,4,14]

Example 2: Input: label = 26 Output: [1,2,6,10,26]

Constraints:

1 <= label <= 10^6

==
Problem Number: 1029 URL: https://leetcode.com/problems/filling-bookcase-shelves Title: 1105. Filling Bookcase Shelves Problem Description: You are given an array books where books[i] = [thicknessi, heighti] indicates the thickness and height of the ith book. You are also given an integer shelfWidth. We want to place these books in order onto bookcase shelves that have a total width shelfWidth. We choose some of the books to place on this shelf such that the sum of their thickness is less than or equal to shelfWidth, then build another level of the shelf of the bookcase so that the total height of the bookcase has increased by the maximum height of the books we just put down. We repeat this process until there are no more books to place. Note that at each step of the above process, the order of the books we place is the same order as the given sequence of books.

For example, if we have an ordered list of 5 books, we might place the first and second book onto the first shelf, the third book on the second shelf, and the fourth and fifth book on the last shelf.

Return the minimum possible height that the total bookshelf can be after placing shelves in this manner.   Example 1:

Input: books = [[1,1],[2,3],[2,3],[1,1],[1,1],[1,1],[1,2]], shelfWidth = 4 Output: 6 Explanation: The sum of the heights of the 3 shelves is 1 + 3 + 2 = 6. Notice that book number 2 does not have to be on the first shelf.

Example 2: Input: books = [[1,3],[2,4],[3,2]], shelfWidth = 6 Output: 4

Constraints:

1 <= books.length <= 1000 1 <= thicknessi <= shelfWidth <= 1000 1 <= heighti <= 1000

==

Problem Number: 1030 URL: https://leetcode.com/problems/corporate-flight-bookings Title: 1109. Corporate Flight Bookings Problem Description: There are n flights that are labeled from 1 to n. You are given an array of flight bookings bookings, where bookings[i] = [firsti, lasti, seatsi] represents a booking for flights firsti through lasti (inclusive) with seatsi seats reserved for each flight in the range. Return an array answer of length n, where answer[i] is the total number of seats reserved for flight i.   Example 1: Input: bookings = [[1,2,10],[2,3,20],[2,5,25]], n = 5 Output: [10,55,45,25,25] Explanation: Flight labels: 1 2 3 4 5 Booking 1 reserved: 10 10 Booking 2 reserved: 20 20 Booking 3 reserved: 25 25 25 25 Total seats: 10 55 45 25 25 Hence, answer = [10,55,45,25,25]

Example 2: Input: bookings = [[1,2,10],[2,2,15]], n = 2 Output: [10,25] Explanation: Flight labels: 1 2 Booking 1 reserved: 10 10 Booking 2 reserved: 15 Total seats: 10 25 Hence, answer = [10,25]

Constraints:

1 <= n <= 2 * 104 1 <= bookings.length <= 2 * 104 bookings[i].length == 3 1 <= firsti <= lasti <= n 1 <= seatsi <= 104

==

Problem Number: 1031 URL: https://leetcode.com/problems/delete-nodes-and-return-forest Title: 1110. Delete Nodes And Return Forest Problem Description: Given the root of a binary tree, each node in the tree has a distinct value. After deleting all nodes with a value in to_delete, we are left with a forest (a disjoint union of trees). Return the roots of the trees in the remaining forest. You may return the result in any order.   Example 1:

Input: root = [1,2,3,4,5,6,7], to_delete = [3,5] Output: [[1,2,null,4],[6],[7]]

Example 2: Input: root = [1,2,4,null,3], to_delete = [3] Output: [[1,2,4]]

Constraints:

The number of nodes in the given tree is at most 1000. Each node has a distinct value between 1 and 1000. to_delete.length <= 1000 to_delete contains distinct values between 1 and 1000.

==
Problem Number: 1032 URL: https://leetcode.com/problems/maximum-nesting-depth-of-two-valid-parentheses-strings Title: 1111. Maximum Nesting Depth of Two Valid Parentheses Strings Problem Description: A string is a valid parentheses string (denoted VPS) if and only if it consists of "(" and ")" characters only, and:

It is the empty string, or It can be written as AB (A concatenated with B), where A and B are VPS's, or It can be written as (A), where A is a VPS.

We can similarly define the nesting depth depth(S) of any VPS S as follows:

depth("") = 0 depth(A + B) = max(depth(A), depth(B)), where A and B are VPS's depth("(" + A + ")") = 1 + depth(A), where A is a VPS.

For example, "", "()()", and "()(()())" are VPS's (with nesting depths 0, 1, and 2), and ")(" and "(()" are not VPS's. Given a VPS seq, split it into two disjoint subsequences A and B, such that A and B are VPS's (and A.length + B.length = seq.length). Now choose any such A and B such that max(depth(A), depth(B)) is the minimum possible value. Return an answer array (of length seq.length) that encodes such a choice of A and B: answer[i] = 0 if seq[i] is part of A, else answer[i] = 1. Note that even though multiple answers may exist, you may return any of them. Example 1: Input: seq = "(()())" Output: [0,1,1,1,1,0]

Example 2: Input: seq = "()(())()" Output: [0,0,0,1,1,0,1,1]

Constraints:

1 <= seq.size <= 10000

==
Problem Number: 1033 URL: https://leetcode.com/problems/lowest-common-ancestor-of-deepest-leaves Title: 1123. Lowest Common Ancestor of Deepest Leaves Problem Description: Given the root of a binary tree, return the lowest common ancestor of its deepest leaves. Recall that:

The node of a binary tree is a leaf if and only if it has no children The depth of the root of the tree is 0. if the depth of a node is d, the depth of each of its children is d + 1. The lowest common ancestor of a set S of nodes, is the node A with the largest depth such that every node in S is in the subtree with root A.

Example 1:

Input: root = [3,5,1,6,2,0,8,null,null,7,4] Output: [2,7,4] Explanation: We return the node with value 2, colored in yellow in the diagram. The nodes coloured in blue are the deepest leaf-nodes of the tree. Note that nodes 6, 0, and 8 are also leaf nodes, but the depth of them is 2, but the depth of nodes 7 and 4 is 3. Example 2: Input: root = [1] Output: [1] Explanation: The root is the deepest node in the tree, and it's the lca of itself.

Example 3: Input: root = [0,1,3,null,2] Output: [2] Explanation: The deepest leaf node in the tree is 2, the lca of one node is itself.

Constraints:

The number of nodes in the tree will be in the range [1, 1000]. 0 <= Node.val <= 1000 The values of the nodes in the tree are unique.

Note: This question is the same as 865: https://leetcode.com/problems/smallest-subtree-with-all-the-deepest-nodes/

==
Problem Number: 1034 URL: https://leetcode.com/problems/longest-well-performing-interval Title: 1124. Longest Well-Performing Interval Problem Description: We are given hours, a list of the number of hours worked per day for a given employee. A day is considered to be a tiring day if and only if the number of hours worked is (strictly) greater than 8. A well-performing interval is an interval of days for which the number of tiring days is strictly larger than the number of non-tiring days. Return the length of the longest well-performing interval. Example 1: Input: hours = [9,9,6,0,6,6,9] Output: 3 Explanation: The longest well-performing interval is [9,9,6].

Example 2: Input: hours = [6,6,6] Output: 0

Constraints:

1 <= hours.length <= 104 0 <= hours[i] <= 16

==
Problem Number: 1035 URL: https://leetcode.com/problems/shortest-path-with-alternating-colors Title: 1129. Shortest Path with Alternating Colors Problem Description: You are given an integer n, the number of nodes in a directed graph where the nodes are labeled from 0 to n - 1. Each edge is red or blue in this graph, and there could be self-edges and parallel edges. You are given two arrays redEdges and blueEdges where:

redEdges[i] = [ai, bi] indicates that there is a directed red edge from node ai to node bi in the graph, and blueEdges[j] = [uj, vj] indicates that there is a directed blue edge from node uj to node vj in the graph.

Return an array answer of length n, where each answer[x] is the length of the shortest path from node 0 to node x such that the edge colors alternate along the path, or -1 if such a path does not exist. Example 1: Input: n = 3, redEdges = [[0,1],[1,2]], blueEdges = [] Output: [0,1,-1]

Example 2: Input: n = 3, redEdges = [[0,1]], blueEdges = [[2,1]] Output: [0,1,-1]

Constraints:

1 <= n <= 100 0 <= redEdges.length, blueEdges.length <= 400 redEdges[i].length == blueEdges[j].length == 2 0 <= ai, bi, uj, vj < n

==
Problem Number: 1036 URL: https://leetcode.com/problems/minimum-cost-tree-from-leaf-values Title: 1130. Minimum Cost Tree From Leaf Values Problem Description: Given an array arr of positive integers, consider all binary trees such that:

Each node has either 0 or 2 children; The values of arr correspond to the values of each leaf in an in-order traversal of the tree. The value of each non-leaf node is equal to the product of the largest leaf value in its left and right subtree, respectively.

Among all possible binary trees considered, return the smallest possible sum of the values of each non-leaf node. It is guaranteed this sum fits into a 32-bit integer. A node is a leaf if and only if it has zero children. Example 1:

Input: arr = [6,2,4] Output: 32 Explanation: There are two possible trees shown. The first has a non-leaf node sum 36, and the second has non-leaf node sum 32.

Example 2:

Input: arr = [4,11] Output: 44

Constraints:

2 <= arr.length <= 40 1 <= arr[i] <= 15 It is guaranteed that the answer fits into a 32-bit signed integer (i.e., it is less than 231).

==
Problem Number: 1037 URL: https://leetcode.com/problems/maximum-of-absolute-value-expression Title: 1131. Maximum of Absolute Value Expression Problem Description: Given two arrays of integers with equal lengths, return the maximum value of: |arr1[i] - arr1[j]| + |arr2[i] - arr2[j]| + |i - j| where the maximum is taken over all 0 <= i, j < arr1.length. Example 1: Input: arr1 = [1,2,3,4], arr2 = [-1,4,5,6] Output: 13

Example 2: Input: arr1 = [1,-2,-5,0,10], arr2 = [0,-2,-1,-7,-4] Output: 20

Constraints:

2 <= arr1.length == arr2.length <= 40000 -10^6 <= arr1[i], arr2[i] <= 10^6

==
Problem Number: 1038 URL: https://leetcode.com/problems/alphabet-board-path Title: 1138. Alphabet Board Path Problem Description: On an alphabet board, we start at position (0, 0), corresponding to character board[0][0]. Here,

board = ["abcde", "fghij", "klmno", "pqrst", "uvwxy", "z"], as shown in the diagram below.

We may make the following moves:

'U' moves our position up one row, if the position exists on the board; 'D' moves our position down one row, if the position exists on the board; 'L' moves our position left one column, if the position exists on the board; 'R' moves our position right one column, if the position exists on the board; '!' adds the character board[r][c] at our current position (r, c) to the answer.

(Here, the only positions that exist on the board are positions with letters on them.) Return a sequence of moves that makes our answer equal to target in the minimum number of moves. You may return any path that does so. Example 1: Input: target = "leet" Output: "DDR!UURRR!!DDD!" Example 2: Input: target = "code" Output: "RR!DDRR!UUL!R!"

Constraints:

1 <= target.length <= 100 target consists only of English lowercase letters.

==

Problem Number: 1039 URL: https://leetcode.com/problems/largest-1-bordered-square Title: 1139. Largest 1-Bordered Square Problem Description: Given a 2D grid of 0s and 1s, return the number of elements in the largest square subgrid that has all 1s on its border, or 0 if such a subgrid doesn't exist in the grid. Example 1: Input: grid = [[1,1,1],[1,0,1],[1,1,1]] Output: 9

Example 2: Input: grid = [[1,1,0,0]] Output: 1

Constraints:

1 <= grid.length <= 100 1 <= grid[0].length <= 100 grid[i][j] is 0 or 1

==

Problem Number: 1040 URL: https://leetcode.com/problems/stone-game-ii Title: 1140. Stone Game II Problem Description: Alice and Bob continue their games with piles of stones. There are a number of piles arranged in a row, and each pile has a positive integer number of stones piles[i]. The objective of the game is to end with the most stones. Alice and Bob take turns, with Alice starting first. Initially, M = 1. On each player's turn, that player can take all the stones in the first X remaining piles, where 1 <= X <= 2M. Then, we set M = max(M, X). The game continues until all the stones have been taken. Assuming Alice and Bob play optimally, return the maximum number of stones Alice can get. Example 1: Input: piles = [2,7,9,4,4] Output: 10 Explanation: If Alice takes one pile at the beginning, Bob takes two piles, then Alice takes 2 piles again. Alice can get 2 + 4 + 4 = 10 piles in total. If Alice takes two piles at the beginning, then Bob can take all three piles left. In this case, Alice get 2 + 7 = 9 piles in total. So we return 10 since it's larger.

Example 2: Input: piles = [1,2,3,4,5,100] Output: 104

Constraints:

1 <= piles.length <= 100 1 <= piles[i] <= 104

===

Problem Number: 1041 URL: https://leetcode.com/problems/longest-common-subsequence Title: 1143. Longest Common Subsequence Problem Description: Given two strings text1 and text2, return the length of their longest common subsequence. If there is no common subsequence, return 0. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, ”ace” is a subsequence of ”abcde”.

A common subsequence of two strings is a subsequence that is common to both strings. Example 1: Input: text1 = ”abcde”, text2 = ”ace” Output: 3 Explanation: The longest common subsequence is ”ace” and its length is 3.

Example 2: Input: text1 = ”abc”, text2 = ”abc” Output: 3 Explanation: The longest common subsequence is ”abc” and its length is 3.

Example 3: Input: text1 = ”abc”, text2 = ”def” Output: 0 Explanation: There is no such common subsequence, so the result is 0.

Constraints:

1 <= text1.length, text2.length <= 1000 text1 and text2 consist of only lowercase English characters.

===

Problem Number: 1042 URL: https://leetcode.com/problems/decrease-elements-to-make-array-zigzag Title: 1144. Decrease Elements To Make Array Zigzag Problem Description: Given an array nums of integers, a move consists of choosing any element and decreasing it by 1. An array A is a zigzag array if either:

Every even-indexed element is greater than adjacent elements, ie. A[0] > A[1] < A[2] > A[3] < A[4] > ... OR, every odd-indexed element is greater than adjacent elements, ie. A[0] < A[1] > A[2] < A[3] > A[4] < ...

Return the minimum number of moves to transform the given array nums into a zigzag array. Example 1: Input: nums = [1,2,3] Output: 2 Explanation: We can decrease 2 to 0 or 3 to 1.

Example 2: Input: nums = [9,6,1,6,2] Output: 4

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 1000

===

Problem Number: 1043 URL: https://leetcode.com/problems/binary-tree-coloring-game Title: 1145. Binary Tree Coloring Game Problem Description: Two players play a turn based game on a binary tree. We are given the root of this binary tree, and the number of nodes n in the tree. n is odd, and each node has a distinct value from 1 to n. Initially, the first player names a value x with 1 <= x <= n, and the second player names a value y with 1 <= y <= n and y != x. The first player colors the node with value x red, and the second player colors the node with value y blue. Then, the players take turns starting with the first player. In each turn, that player chooses a node of their color (red if player 1, blue if player 2) and colors an uncolored neighbor of the chosen node (either the left child, right child, or parent of the chosen node.) If (and only if) a player cannot choose such a node in this way, they must pass their turn. If both players pass their turn, the game ends, and the winner is the player that colored more nodes. You are the second player. If it is possible to choose such a y to ensure you win the game, return true. If it is not possible, return false. Example 1:

Input: root = [1,2,3,4,5,6,7,8,9,10,11], n = 11, x = 3 Output: true Explanation: The second player can choose the node with value 2.

Example 2: Input: root = [1,2,3], n = 3, x = 1 Output: false

Constraints:

The number of nodes in the tree is n. 1 <= x <= n <= 100 n is odd. 1 <= Node.val <= n All the values of the tree are unique.

==

Problem Number: 1044 URL: https://leetcode.com/problems/snapshot-array Title: 1146. Snapshot Array Problem Description: Implement a SnapshotArray that supports the following interface:

SnapshotArray(int length) initializes an array-like data structure with the given length. Initially, each element equals 0. void set(index, val) sets the element at the given index to be equal to val. int snap() takes a snapshot of the array and returns the snap_id: the total number of times we called snap() minus 1. int get(index, snap_id) returns the value at the given index, at the time we took the snapshot with the given snap_id

Example 1: Input: ["SnapshotArray","set","snap","set","get"] [[3],[0,5],[],[0,6],[0,0]] Output: [null,null,0,null,5] Explanation: SnapshotArray snapshotArr = new SnapshotArray(3); // set the length to be 3 snapshotArr.set(0,5); // Set array[0] = 5 snapshotArr.snap(); // Take a snapshot, return snap_id = 0 snapshotArr.set(0,6); snapshotArr.get(0,0); // Get the value of array[0] with snap_id = 0, return 5 Constraints:

1 <= length <= 5 * 104 0 <= index < length 0 <= val <= 109 0 <= snap_id < (the total number of times we call snap()) At most 5 * 104 calls will be made to set, snap, and get.

==

Problem Number: 1045 URL: https://leetcode.com/problems/number-of-dice-rolls-with-target-sum Title: 1155. Number of Dice Rolls With Target Sum Problem Description: You have n dice, and each die has k faces numbered from 1 to k. Given three integers n, k, and target, return the number of possible ways (out of the kn total ways) to roll the dice, so the sum of the face-up numbers equals target. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: n = 1, k = 6, target = 3 Output: 1 Explanation: You throw one die with 6 faces. There is only one way to get a sum of 3.

Example 2: Input: n = 2, k = 6, target = 7 Output: 6 Explanation: You throw two dice, each with 6 faces. There are 6 ways to get a sum of 7: 1+6, 2+5, 3+4, 4+3, 5+2, 6+1.

Example 3: Input: n = 30, k = 30, target = 500 Output: 222616187 Explanation: The answer must be returned modulo 109 + 7.

Constraints:

1 <= n, k <= 30 1 <= target <= 1000

==
Problem Number: 1046 URL: https://leetcode.com/problems/swap-for-longest-repeated-character-substring Title: 1156. Swap For Longest Repeated Character Substring Problem Description: You are given a string text. You can swap two of the characters in the text. Return the length of the longest substring with repeated characters. Example 1: Input: text = "ababa" Output: 3 Explanation: We can swap the first 'b' with the last 'a', or the last 'b' with the first 'a'. Then, the longest repeated character substring is "aaa" with length 3.

Example 2: Input: text = "aaabaaa" Output: 6 Explanation: Swap 'b' with the last 'a' (or the first 'a'), and we get longest repeated character substring "aaaaaa" with length 6.

Example 3: Input: text = "aaaaa" Output: 5 Explanation: No need to swap, longest repeated character substring is "aaaaa" with length is 5.

Constraints:

1 <= text.length <= 2 * 104 text consist of lowercase English characters only.

==
Problem Number: 1047 URL: https://leetcode.com/problems/maximum-level-sum-of-a-binary-tree Title: 1161. Maximum Level Sum of a Binary Tree Problem Description: Given the root of a binary tree, the level of its root is 1, the level of its children is 2, and so on. Return the smallest level x such that the sum of all the values of nodes at level x is maximal. Example 1:

Input: root = [1,7,0,7,-8,null,null] Output: 2 Explanation: Level 1 sum = 1. Level 2 sum = 7 + 0 = 7. Level 3 sum = 7 + -8 = -1. So we return the level with the maximum sum which is level 2.

Example 2: Input: root = [989,null,10250,98693,-89388,null,null,null,-32127] Output: 2

Constraints:

The number of nodes in the tree is in the range [1, 104]. -105 <= Node.val <= 105

==

Problem Number: 1048 URL: https://leetcode.com/problems/as-far-from-land-as-possible Title: 1162. As Far from Land as Possible Problem Description: Given an n x n grid containing only values 0 and 1, where 0 represents water and 1 represents land, find a water cell such that its distance to the nearest land cell is maximized, and return the distance. If no land or water exists in the grid, return -1. The distance used in this problem is the Manhattan distance: the distance between two cells (x0, y0) and (x1, y1) is |x0 - x1| + |y0 - y1|. Example 1:

Input: grid = [[1,0,1],[0,0,0],[1,0,1]] Output: 2 Explanation: The cell (1, 1) is as far as possible from all the land with distance 2.

Example 2:

Input: grid = [[1,0,0],[0,0,0],[0,0,0]] Output: 4 Explanation: The cell (2, 2) is as far as possible from all the land with distance 4.

Constraints:

n == grid.length n == grid[i].length 1 <= n <= 100 grid[i][j] is 0 or 1

==

Problem Number: 1049 URL: https://leetcode.com/problems/invalid-transactions Title: 1169. Invalid Transactions Problem Description: A transaction is possibly invalid if:

the amount exceeds $1000, or; if it occurs within (and including) 60 minutes of another transaction with the same name in a different city.

You are given an array of strings transaction where transactions[i] consists of comma-separated values representing the name, time (in minutes), amount, and city of the transaction. Return a list of transactions that are possibly invalid. You may return the answer in any order. Example 1: Input: transactions = ["alice,20,800,mtv","alice,50,100,beijing"] Output: ["alice,20,800,mtv","alice,50,100,beijing"] Explanation: The first transaction is invalid because the second transaction occurs within a difference of 60 minutes, have the same name and is in a different city. Similarly the second one is invalid too. Example 2: Input: transactions = ["alice,20,800,mtv","alice,50,1200,mtv"] Output: ["alice,50,1200,mtv"]

Example 3: Input: transactions = ["alice,20,800,mtv","bob,50,1200,mtv"] Output: ["bob,50,1200,mtv"]

Constraints:

transactions.length <= 1000 Each transactions[i] takes the form "{name},{time},{amount},{city}" Each {name} and {city} consist of lowercase English letters, and have lengths between 1 and 10. Each {time} consist of digits, and represent an integer between 0 and 1000. Each {amount} consist of digits, and represent an integer between 0 and 2000.

==
Problem Number: 1050 URL: https://leetcode.com/problems/compare-strings-by-frequency-of-the-smallest-character Title: 1170. Compare Strings by Frequency of the Smallest Character Problem Description: Let the function f(s) be the frequency of the lexicographically smallest character in a non-empty string s. For example, if s = "dcce" then f(s) = 2 because the lexicographically smallest character is 'c', which has a frequency of 2. You are given an array of strings words and another array of query strings queries. For each query queries[i], count the number of words in words such that f(queries[i]) < f(W) for each W in words. Return an integer array answer, where each answer[i] is the answer to the ith query. Example 1: Input: queries = ["cbd"], words = ["zaaaz"] Output: [1] Explanation: On the first query we have f("cbd") = 1, f("zaaaz") = 3 so f("cbd") < f("zaaaz").

Example 2: Input: queries = ["bbb","cc"], words = ["a","aa","aaa","aaaa"] Output: [1,2] Explanation: On the first query only f("bbb") < f("aaaa"). On the second query both f("aaa") and f("aaaa") are both > f("cc").

Constraints:

1 <= queries.length <= 2000 1 <= words.length <= 2000 1 <= queries[i].length, words[i].length <= 10 queries[i][j], words[i][j] consist of lowercase English letters.

==
Problem Number: 1051 URL: https://leetcode.com/problems/remove-zero-sum-consecutive-nodes-from-linked-list Title: 1171. Remove Zero Sum Consecutive Nodes from Linked List Problem Description: Given the head of a linked list, we repeatedly delete consecutive sequences of nodes that sum to 0 until there are no such sequences. After doing so, return the head of the final linked list. You may return any such answer. (Note that in the examples below, all sequences are serializations of ListNode objects.) Example 1: Input: head = [1,2,-3,3,1] Output: [3,1] Note: The answer [1,2,1] would also be accepted.

Example 2: Input: head = [1,2,3,-3,4] Output: [1,2,4]

Example 3: Input: head = [1,2,3,-3,-2] Output: [1]

Constraints:

The given linked list will contain between 1 and 1000 nodes. Each node in the linked list has -1000 <= node.val <= 1000.

==
Problem Number: 1052 URL: https://leetcode.com/problems/can-make-palindrome-from-substring Title: 1177. Can Make Palindrome from Substring Problem Description: You are given a string s and array queries where queries[i] = [lefti, righti, ki]. We may rearrange the substring s[lefti...righti] for each query and then choose up to ki of them to replace with any lowercase English letter. If the substring is possible to be a palindrome string after the operations above, the result of the query is true. Otherwise, the result is false. Return a boolean array answer where answer[i] is the result of the ith query queries[i]. Note that each letter is counted individually for replacement, so if, for example s[lefti...righti] = "aaa", and ki = 2, we can only replace two of the letters. Also, note that no query modifies the initial string s. Example : Input: s = "abcda", queries = [[3,3,0],[1,2,0],[0,3,1],[0,3,2],[0,4,1]] Output: [true,false,false,true,true] Explanation: queries[0]: substring = "d", is palidrome. queries[1]: substring = "bc", is not palidrome. queries[2]: substring = "abcd", is not palidrome after replacing only 1 character. queries[3]: substring = "abcd", could be changed to "abba" which is palidrome. Also this can be changed to "baab" first rearrange it "bacd" then replace "cd" with "ab". queries[4]: substring = "abcda", could be changed to "abcba" which is palidrome.

Example 2: Input: s = "lyb", queries = [[0,1,0],[2,2,1]] Output: [false,true]

Constraints:

1 <= s.length, queries.length <= 105 0 <= lefti <= righti < s.length 0 <= ki <= s.length s consists of lowercase English letters.

==
Problem Number: 1053 URL: https://leetcode.com/problems/maximum-subarray-sum-with-one-deletion Title: 1186. Maximum Subarray Sum with One Deletion Problem Description: Given an array of integers, return the maximum sum for a non-empty subarray (contiguous elements) with at most one element deletion. In other words, you want to choose a subarray and optionally delete one element from it so that there is still at least one element left and the sum of the remaining elements is maximum possible. Note that the subarray needs to be non-empty after deleting one element. Example 1: Input: arr = [1,-2,0,3] Output: 4 Explanation: Because we can choose [1, -2, 0, 3] and drop -2, thus the subarray [1, 0, 3] becomes the maximum value. Example 2: Input: arr = [1,-2,-2,3] Output: 3 Explanation: We just choose [3] and it's the maximum sum.

Example 3: Input: arr = [-1,-1,-1,-1] Output: -1 Explanation: The final subarray needs to be non-empty. You can't choose [-1] and delete -1 from it, then get an empty subarray to make the sum equals to 0.

Constraints:

1 <= arr.length <= 105 -104 <= arr[i] <= 104

==

Problem Number: 1054 URL: https://leetcode.com/problems/reverse-substrings-between-each-pair-of-parentheses Title: 1190. Reverse Substrings Between Each Pair of Parentheses Problem Description: You are given a string s that consists of lower case English letters and brackets. Reverse the strings in each pair of matching parentheses, starting from the innermost one. Your result should not contain any brackets. Example 1: Input: s = "(abcd)" Output: "dcba"

Example 2: Input: s = "(u(love)i)" Output: "iloveu" Explanation: The substring "love" is reversed first, then the whole string is reversed.

Example 3: Input: s = "(ed(et(oc))el)" Output: "leetcode" Explanation: First, we reverse the substring "oc", then "etco", and finally, the whole string.

Constraints:

1 <= s.length <= 2000 s only contains lower case English characters and parentheses. It is guaranteed that all parentheses are balanced.

==

Problem Number: 1055 URL: https://leetcode.com/problems/k-concatenation-maximum-sum Title: 1191. K-Concatenation Maximum Sum Problem Description: Given an integer array arr and an integer k, modify the array by repeating it k times. For example, if arr = [1, 2] and k = 3 then the modified array will be [1, 2, 1, 2, 1, 2]. Return the maximum sub-array sum in the modified array. Note that the length of the sub-array can be 0 and its sum in that case is 0. As the answer can be very large, return the answer modulo 109 + 7. Example 1: Input: arr = [1,2], k = 3 Output: 9

Example 2: Input: arr = [1,-2,1], k = 5 Output: 2

Example 3: Input: arr = [-1,-2], k = 7 Output: 0

Constraints:

1 <= arr.length <= 105 1 <= k <= 105 -104 <= arr[i] <= 104

==

Problem Number: 1056 URL: https://leetcode.com/problems/ugly-number-iii Title: 1201. Ugly Number III Problem Description: An ugly number is a positive integer that is divisible by a, b, or c. Given four integers n, a, b, and c, return the nth ugly number. Example 1: Input: n = 3, a = 2, b = 3, c = 5 Output: 4 Explanation: The ugly numbers are 2, 3, 4, 5, 6, 8, 9, 10... The 3rd is 4.

Example 2: Input: n = 4, a = 2, b = 3, c = 4 Output: 6 Explanation: The ugly numbers are 2, 3, 4, 6, 8, 9, 10, 12... The 4th is 6.

Example 3: Input: n = 5, a = 2, b = 11, c = 13 Output: 10 Explanation: The ugly numbers are 2, 4, 6, 8, 10, 11, 12, 13... The 5th is 10.

Constraints:

1 <= n, a, b, c <= 109 1 <= a * b * c <= 1018 It is guaranteed that the result will be in range [1, 2 * 109].

===
Problem Number: 1057 URL: https://leetcode.com/problems/smallest-string-with-swaps Title: 1202. Smallest String With Swaps Problem Description: You are given a string s, and an array of pairs of indices in the string pairs where pairs[i] = [a, b] indicates 2 indices(0-indexed) of the string. You can swap the characters at any pair of indices in the given pairs any number of times. Return the lexicographically smallest string that s can be changed to after using the swaps. Example 1: Input: s = "dcab", pairs = [[0,3],[1,2]] Output: "bacd" Explaination: Swap s[0] and s[3], s = "bcad" Swap s[1] and s[2], s = "bacd"

Example 2: Input: s = "dcab", pairs = [[0,3],[1,2],[0,2]] Output: "abcd" Explaination: Swap s[0] and s[3], s = "bcad" Swap s[0] and s[2], s = "acbd" Swap s[1] and s[2], s = "abcd" Example 3: Input: s = "cba", pairs = [[0,1],[1,2]] Output: "abc" Explaination: Swap s[0] and s[1], s = "bca" Swap s[1] and s[2], s = "bac" Swap s[0] and s[1], s = "abc"

Constraints:

1 <= s.length <= 10^5 0 <= pairs.length <= 10^5 0 <= pairs[i][0], pairs[i][1] < s.length s only contains lower case English letters.

===
Problem Number: 1058 URL: https://leetcode.com/problems/get-equal-substrings-within-budget Title: 1208. Get Equal Substrings Within Budget Problem Description: You are given two strings s and t of the same length and an integer maxCost. You want to change s to t. Changing the ith character of s to ith character of t costs |s[i] - t[i]| (i.e., the absolute difference between the ASCII values of the characters). Return the maximum length of a substring of s that can be changed to be the same as the corresponding substring of t with a cost less than or equal to maxCost. If there is no substring from s that can be changed to its corresponding substring from t, return 0. Example 1: Input: s = "abcd", t = "bcdf", maxCost = 3 Output: 3 Explanation: "abc" of s can change to "bcd". That costs 3, so the maximum length is 3.

Example 2: Input: s = "abcd", t = "cdef", maxCost = 3 Output: 1 Explanation: Each character in s costs 2 to change to character in t, so the maximum length is 1.

Example 3: Input: s = "abcd", t = "acde", maxCost = 0 Output: 1 Explanation: You cannot make any change, so the maximum length is 1.

Constraints:

1 <= s.length <= 105 t.length == s.length 0 <= maxCost <= 106 s and t consist of only lowercase English letters.

==
Problem Number: 1059 URL: https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string-ii Title: 1209. Remove All Adjacent Duplicates in String II Problem Description: You are given a string s and an integer k, a k duplicate removal consists of choosing k adjacent and equal letters from s and removing them, causing the left and the right side of the deleted substring to concatenate together. We repeatedly make k duplicate removals on s until we no longer can. Return the final string after all such duplicate removals have been made. It is guaranteed that the answer is unique. Example 1: Input: s = "abcd", k = 2 Output: "abcd" Explanation: There's nothing to delete. Example 2: Input: s = "deeedbbcccbdaa", k = 3 Output: "aa" Explanation: First delete "eee" and "ccc", get "ddbbbdaa" Then delete "bbb", get "dddaa" Finally delete "ddd", get "aa" Example 3: Input: s = "pbbcggttciiippooaais", k = 2 Output: "ps"

Constraints:

1 <= s.length <= 105 2 <= k <= 104 s only contains lowercase English letters.

==
Problem Number: 1060 URL: https://leetcode.com/problems/longest-arithmetic-subsequence-of-given-difference Title: 1218. Longest Arithmetic Subsequence of Given Difference Problem Description: Given an integer array arr and an integer difference, return the length of the longest subsequence in arr which is an arithmetic sequence such that the difference between adjacent elements in the subsequence equals difference. A subsequence is a sequence that can be derived from arr by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: arr = [1,2,3,4], difference = 1 Output: 4 Explanation: The longest arithmetic subsequence is [1,2,3,4]. Example 2: Input: arr = [1,3,5,7], difference = 1 Output: 1 Explanation: The longest arithmetic subsequence is any single element.

Example 3: Input: arr = [1,5,7,8,5,3,4,2,1], difference = -2 Output: 4 Explanation: The longest arithmetic subsequence is [7,5,3,1].

Constraints:

1 <= arr.length <= 105 -104 <= arr[i], difference <= 104

==
Problem Number: 1061 URL: https://leetcode.com/problems/path-with-maximum-gold Title: 1219. Path with Maximum Gold Problem Description: In a gold mine grid of size m x n, each cell in this mine has an integer representing the amount of gold in that cell, 0 if it is empty. Return the maximum amount of gold you can collect under the conditions:

Every time you are located in a cell you will collect all the gold in that cell. From your position, you can walk one step to the left, right, up, or down. You can't visit the same cell more than once. Never visit a cell with 0 gold. You can start and stop collecting gold from any position in the grid that has some gold.

Example 1: Input: grid = [[0,6,0],[5,8,7],[0,9,0]] Output: 24 Explanation: [[0,6,0], [5,8,7], [0,9,0]] Path to get the maximum gold, 9 -> 8 -> 7.

Example 2: Input: grid = [[1,0,7],[2,0,6],[3,4,5],[0,3,0],[9,0,20]] Output: 28 Explanation: [[1,0,7], [2,0,6], [3,4,5], [0,3,0], [9,0,20]] Path to get the maximum gold, 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 15 0 <= grid[i][j] <= 100 There are at most 25 cells containing gold.

==
Problem Number: 1062 URL: https://leetcode.com/problems/queens-that-can-attack-the-king Title: 1222. Queens That Can Attack the King Problem Description: On a 0-indexed 8 x 8 chessboard, there can be multiple black queens ad one white king. You are given a 2D integer array queens where queens[i] = [xQueeni, yQueeni] represents the position of the ith black queen on the chessboard. You are also given an integer array king of length 2 where king = [xKing, yKing] represents the position of the white king. Return the coordinates of the black queens that can directly attack the king. You may return the answer in any order. Example 1:

Input: queens = [[0,1],[1,0],[4,0],[0,4],[3,3],[2,4]], king = [0,0] Output: [[0,1],[1,0],[3,3]] Explanation: The diagram above shows the three queens that can directly attack the king and the three queens that cannot attack the king (i.e., marked with red dashes).

Example 2:

Input: queens = [[0,0],[1,1],[2,2],[3,4],[3,5],[4,4],[4,5]], king = [3,3] Output: [[2,2],[3,4],[4,4]] Explanation: The diagram above shows the three queens that can directly attack the king and the three queens that cannot attack the king (i.e., marked with red dashes).

Constraints:

1 <= queens.length < 64 queens[i].length == king.length == 2 0 <= xQueeni, yQueeni, xKing, yKing < 8 All the given positions are unique.

==
Problem Number: 1063 URL: https://leetcode.com/problems/airplane-seat-assignment-probability Title: 1227. Airplane Seat Assignment Probability Problem Description: n passengers board an airplane with exactly n seats. The first passenger has lost the ticket and picks a seat randomly. But after that, the rest of the passengers will:

Take their own seat if it is still available, and Pick other seats randomly when they find their seat occupied

Return the probability that the nth person gets his own seat. Example 1: Input: n = 1 Output: 1.00000 Explanation: The first person can only get the first seat. Example 2: Input: n = 2 Output: 0.50000 Explanation: The second person has a probability of 0.5 to get the second seat (when first person gets the first seat).

Constraints:

1 <= n <= 105

==

Problem Number: 1064 URL: https://leetcode.com/problems/remove-sub-folders-from-the-filesystem Title: 1233. Remove Sub-Folders from the Filesystem Problem Description: Given a list of folders folder, return the folders after removing all sub-folders in those folders. You may return the answer in any order. If a folder[i] is located within another folder[j], it is called a sub-folder of it. The format of a path is one or more concatenated strings of the form: '/' followed by one or more lowercase English letters.

For example, "/leetcode" and "/leetcode/problems" are valid paths while an empty string and "/" are not.

Example 1: Input: folder = ["/a","/a/b","/c/d","/c/d/e","/c/f"] Output: ["/a","/c/d","/c/f"] Explanation: Folders "/a/b" is a subfolder of "/a" and "/c/d/e" is inside of folder "/c/d" in our filesystem.

Example 2: Input: folder = ["/a","/a/b/c","/a/b/d"] Output: ["/a"] Explanation: Folders "/a/b/c" and "/a/b/d" will be removed because they are subfolders of "/a".

Example 3: Input: folder = ["/a/b/c","/a/b/ca","/a/b/d"] Output: ["/a/b/c","/a/b/ca","/a/b/d"]

Constraints:

1 <= folder.length <= 4 * 104 2 <= folder[i].length <= 100 folder[i] contains only lowercase letters and '/'. folder[i] always starts with the character '/'. Each folder name is unique.

==

Problem Number: 1065 URL: https://leetcode.com/problems/replace-the-substring-for-balanced-string Title: 1234. Replace the Substring for Balanced String Problem Description: You are given a string s of length n containing only four kinds of characters: 'Q', 'W', 'E', and 'R'. A string is said to be balanced if each of its characters appears n / 4 times where n is the length of the string. Return the minimum length of the substring that can be replaced with any other string of the same length to make s balanced. If s is already balanced, return 0. Example 1: Input: s = "QWER" Output: 0 Explanation: s is already balanced.

Example 2: Input: s = "QQWE" Output: 1 Explanation: We need to replace a 'Q' to 'R', so that "RQWE" (or "QRWE") is balanced.

Example 3: Input: s = "QQQW" Output: 2 Explanation: We can replace the first "QQ" to "ER".

Constraints:

n == s.length 4 <= n <= 105 n is a multiple of 4. s contains only 'Q', 'W', 'E', and 'R'.

==

Problem Number: 1066 URL: https://leetcode.com/problems/find-positive-integer-solution-for-a-given-equation Title: 1237. Find Positive Integer Solution for a Given Equation Problem Description: Given a callable function f(x, y) with a hidden formula and a value z, reverse engineer the formula and return all positive integer pairs x and y where f(x,y) == z. You may return the pairs in any order. While the exact formula is hidden, the function is monotonically increasing, i.e.:

f(x, y) < f(x + 1, y) f(x, y) < f(x, y + 1)

The function interface is defined like this: interface CustomFunction { public: // Returns some positive integer f(x, y) for two positive integers x and y based on a formula. int f(int x, int y); };

We will judge your solution as follows:

The judge has a list of 9 hidden implementations of CustomFunction, along with a way to generate an answer key of all valid pairs for a specific z. The judge will receive two inputs: a function_id (to determine which implementation to test your code with), and the target z. The judge will call your findSolution and compare your results with the answer key. If your results match the answer key, your solution will be Accepted.

Example 1: Input: function_id = 1, z = 5 Output: [[1,4],[2,3],[3,2],[4,1]] Explanation: The hidden formula for function_id = 1 is f(x, y) = x + y. The following positive integer values of x and y make f(x, y) equal to 5: x=1, y=4 -> f(1, 4) = 1 + 4 = 5. x=2, y=3 -> f(2, 3) = 2 + 3 = 5. x=3, y=2 -> f(3, 2) = 3 + 2 = 5. x=4, y=1 -> f(4, 1) = 4 + 1 = 5.

Example 2: Input: function_id = 2, z = 5 Output: [[1,5],[5,1]] Explanation: The hidden formula for function_id = 2 is f(x, y) = x * y. The following positive integer values of x and y make f(x, y) equal to 5: x=1, y=5 -> f(1, 5) = 1 * 5 = 5. x=5, y=1 -> f(5, 1) = 5 * 1 = 5.

Constraints:

1 <= function_id <= 9 1 <= z <= 100 It is guaranteed that the solutions of f(x, y) == z will be in the range 1 <= x, y <= 1000. It is also guaranteed that f(x, y) will fit in 32 bit signed integer if 1 <= x, y <= 1000.

==

Problem Number: 1067 URL: https://leetcode.com/problems/circular-permutation-in-binary-representation Title: 1238. Circular Permutation in

Binary Representation Problem Description: Given 2 integers n and start. Your task is return any permutation p of (0,1,2.....,2^n -1) such that :

p[0] = start p[i] and p[i+1] differ by only one bit in their binary representation. p[0] and p[2^n -1] must also differ by only one bit in their binary representation.

Example 1: Input: n = 2, start = 3 Output: [3,2,0,1] Explanation: The binary representation of the permutation is (11,10,00,01). All the adjacent element differ by one bit. Another valid permutation is [3,1,0,2]

Example 2: Input: n = 3, start = 2 Output: [2,6,7,5,4,0,1,3] Explanation: The binary representation of the permutation is (010,110,111,101,100,000,001,011).

Constraints:

1 <= n <= 16 0 <= start < 2 ^ n

==
Problem Number: 1068 URL: https://leetcode.com/problems/maximum-length-of-a-concatenated-string-with-unique-characters Title: 1239. Maximum Length of a Concatenated String with Unique Characters Problem Description: You are given an array of strings arr. A string s is formed by the concatenation of a subsequence of arr that has unique characters. Return the maximum possible length of s. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: arr = ["un","iq","ue"] Output: 4 Explanation: All the valid concatenations are: - "" - "un" - "iq" - "ue" - "uniq" ("un" + "iq") - "ique" ("iq" + "ue") Maximum length is 4.

Example 2: Input: arr = ["cha","r","act","ers"] Output: 6 Explanation: Possible longest valid concatenations are "chaers" ("cha" + "ers") and "acters" ("act" + "ers").

Example 3: Input: arr = ["abcdefghijklmnopqrstuvwxyz"] Output: 26 Explanation: The only string in arr has all 26 characters.

Constraints:

1 <= arr.length <= 16 1 <= arr[i].length <= 26 arr[i] contains only lowercase English letters.

==
Problem Number: 1069 URL: https://leetcode.com/problems/minimum-swaps-to-make-strings-equal Title: 1247. Minimum Swaps to Make Strings Equal Problem Description: You are given two strings s1 and s2 of equal length consisting of letters "x" and "y" only. Your task is to make these two strings equal to each other. You can swap any two characters that belong to different strings, which means: swap s1[i] and s2[j]. Return the minimum number of swaps required to make s1 and s2 equal, or return -1 if it is impossible to do so. Example 1: Input: s1 = "xx", s2 = "yy" Output: 1 Explanation: Swap s1[0] and s2[1], s1 = "yx", s2 = "yx".

Example 2: Input: s1 = "xy", s2 = "yx" Output: 2 Explanation: Swap s1[0] and s2[0], s1 = "yy", s2 = "xx". Swap s1[0] and s2[1], s1 = "xy", s2 = "xy". Note that you cannot swap s1[0] and s1[1] to make s1 equal to "yx", cause we can only swap chars in different strings.

Example 3: Input: s1 = "xx", s2 = "xy" Output: -1

Constraints:

1 <= s1.length, s2.length <= 1000 s1.length == s2.length s1, s2 only contain 'x' or 'y'.

==

Problem Number: 1070 URL: https://leetcode.com/problems/count-number-of-nice-subarrays Title: 1248. Count Number of Nice Subarrays Problem Description: Given an array of integers nums and an integer k. A continuous subarray is called nice if there are k odd numbers on it. Return the number of nice sub-arrays. Example 1: Input: nums = [1,1,2,1,1], k = 3 Output: 2 Explanation: The only sub-arrays with 3 odd numbers are [1,1,2,1] and [1,2,1,1].

Example 2: Input: nums = [2,4,6], k = 1 Output: 0 Explanation: There is no odd numbers in the array.

Example 3: Input: nums = [2,2,2,1,2,2,1,2,2,2], k = 2 Output: 16

Constraints:

1 <= nums.length <= 50000 1 <= nums[i] <= 10^5 1 <= k <= nums.length

==

Problem Number: 1071 URL: https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses Title: 1249. Minimum Remove to Make Valid Parentheses Problem Description: Given a string s of '(' , ')' and lowercase English characters. Your task is to remove the minimum number of parentheses ('(' or ')', in any positions) so that the resulting parentheses string is valid and return any valid string. Formally, a parentheses string is valid if and only if:

It is the empty string, contains only lowercase characters, or It can be written as AB (A concatenated with B), where A and B are valid strings, or It can be written as (A), where A is a valid string.

Example 1: Input: s = "lee(t(c)o)de)" Output: "lee(t(c)o)de" Explanation: "lee(t(co)de)" , "lee(t(c)ode)" would also be accepted.

Example 2: Input: s = "a)b(c)d" Output: "ab(c)d"

Example 3: Input: s = "))((" Output: "" Explanation: An empty string is also valid.

Constraints:

1 <= s.length <= 105 s[i] is either'(' , ')', or lowercase English letter.

==
Problem Number: 1072 URL: https://leetcode.com/problems/reconstruct-a-2-row-binary-matrix Title: 1253. Reconstruct a 2-Row Binary Matrix Problem Description: Given the following details of a matrix with n columns and 2 rows :

The matrix is a binary matrix, which means each element in the matrix can be 0 or 1. The sum of elements of the 0-th(upper) row is given as upper. The sum of elements of the 1-st(lower) row is given as lower. The sum of elements in the i-th column(0-indexed) is colsum[i], where colsum is given as an integer array with length n.

Your task is to reconstruct the matrix with upper, lower and colsum. Return it as a 2-D integer array. If there are more than one valid solution, any of them will be accepted. If no valid solution exists, return an empty 2-D array. Example 1: Input: upper = 2, lower = 1, colsum = [1,1,1] Output: [[1,1,0],[0,0,1]] Explanation: [[1,0,1],[0,1,0]], and [[0,1,1],[1,0,0]] are also correct answers.

Example 2: Input: upper = 2, lower = 3, colsum = [2,2,1,1] Output: []

Example 3: Input: upper = 5, lower = 5, colsum = [2,1,2,0,1,0,1,2,0,1] Output: [[1,1,1,0,1,0,0,1,0,0],[1,0,1,0,0,0,1,1,0,1]]

Constraints:

1 <= colsum.length <= 10^5 0 <= upper, lower <= colsum.length 0 <= colsum[i] <= 2

==
Problem Number: 1073 URL: https://leetcode.com/problems/number-of-closed-islands Title: 1254. Number of Closed Islands Problem Description: Given a 2D grid consists of 0s (land) and 1s (water). An island is a maximal 4-directionally connected group of 0s and a closed island is an island totally (all left, top, right, bottom) surrounded by 1s. Return the number of closed islands. Example 1:

Input: grid = [[1,1,1,1,1,1,1,0],[1,0,0,0,0,1,1,0],[1,0,1,0,1,1,1,0],[1,0,0,0,0,1,0,1],[1,1,1,1,1,1,1,0]] Output: 2 Explanation: Islands in gray are closed because they are completely surrounded by water (group of 1s). Example 2:

Input: grid = [[0,0,1,0,0],[0,1,0,1,0],[0,1,1,1,0]] Output: 1

Example 3: Input: grid = [[1,1,1,1,1,1,1], [1,0,0,0,0,0,1], [1,0,1,1,1,0,1], [1,0,1,0,1,0,1], [1,0,1,1,1,0,1], [1,0,0,0,0,0,1], [1,1,1,1,1,1,1]] Output: 2

Constraints:

1 <= grid.length, grid[0].length <= 100 0 <= grid[i][j] <=1

==
Problem Number: 1074 URL: https://leetcode.com/problems/find-elements-

in-a-contaminated-binary-tree Title: 1261. Find Elements in a Contaminated Binary Tree Problem Description: Given a binary tree with the following rules:

root.val == 0 If treeNode.val == x and treeNode.left != null, then treeNode.left.val == 2 * x + 1 If treeNode.val == x and treeNode.right != null, then treeNode.right.val == 2 * x + 2

Now the binary tree is contaminated, which means all treeNode.val have been changed to -1. Implement the FindElements class:

FindElements(TreeNode* root) Initializes the object with a contaminated binary tree and recovers it. bool find(int target) Returns true if the target value exists in the recovered binary tree.

Example 1:

Input ["FindElements","find","find"] [[[-1,null,-1]],[1],[2]] Output [null,false,true] Explanation FindElements findElements = new FindElements([-1,null,-1]); findElements.find(1); // return False findElements.find(2); // return True Example 2:

Input ["FindElements","find","find","find"] [[[-1,-1,-1,-1,-1]],[1],[3],[5]] Output [null,true,true,false] Explanation FindElements findElements = new FindElements([-1,-1,-1,-1,-1]); findElements.find(1); // return True findElements.find(3); // return True findElements.find(5); // return False Example 3:

Input ["FindElements","find","find","find","find"] [[[-1,null,-1,-1,null,-1]],[2],[3],[4],[5]] Output [null,true,false,false,true] Explanation FindElements findElements = new FindElements([-1,null,-1,-1,null,-1]); findElements.find(2); // return True findElements.find(3); // return False findElements.find(4); // return False findElements.find(5); // return True

Constraints:

TreeNode.val == -1 The height of the binary tree is less than or equal to 20 The total number of nodes is between [1, 104] Total calls of find() is between [1, 104] 0 <= target <= 106

==
Problem Number: 1075 URL: https://leetcode.com/problems/greatest-sum-divisible-by-three Title: 1262. Greatest Sum Divisible by Three Problem Description: Given an integer array nums, return the maximum possible sum of elements of the array such that it is divisible by three. Example 1: Input: nums = [3,6,5,1,8] Output: 18 Explanation: Pick numbers 3, 6, 1 and 8 their sum is 18 (maximum sum divisible by 3). Example 2: Input: nums = [4] Output: 0 Explanation: Since 4 is not divisible by 3, do not pick any number.

Example 3: Input: nums = [1,2,3,4,4] Output: 12 Explanation: Pick numbers 1, 3, 4 and 4 their sum is 12 (maximum sum divisible by 3).

Constraints:

1 <= nums.length <= 4 * 104 1 <= nums[i] <= 104

==
Problem Number: 1076 URL: https://leetcode.com/problems/count-servers-that-communicate Title: 1267. Count Servers that Communicate Problem Description: You are given a map of a server center, represented as a m * n integer matrix grid, where 1 means that on that cell there is a server and 0 means that it is no server. Two servers are said to communicate if they are on the same row or on the same column.

Return the number of servers that communicate with any other server. Example 1:

Input: grid = [[1,0],[0,1]] Output: 0 Explanation: No servers can communicate with others. Example 2:

Input: grid = [[1,0],[1,1]] Output: 3 Explanation: All three servers can communicate with at least one other server.

Example 3:

Input: grid = [[1,1,0,0],[0,0,1,0],[0,0,1,0],[0,0,0,1]] Output: 4 Explanation: The two servers in the first row can communicate with each other. The two servers in the third column can communicate with each other. The server at right bottom corner can't communicate with any other server.

Constraints:

m == grid.length n == grid[i].length 1 <= m <= 250 1 <= n <= 250 grid[i][j] == 0 or 1

==
Problem Number: 1077 URL: https://leetcode.com/problems/search-suggestions-system Title: 1268. Search Suggestions System Problem Description: You are given an array of strings products and a string searchWord. Design a system that suggests at most three product names from products after each character of searchWord is typed. Suggested products should have common prefix with searchWord. If there are more than three products with a common prefix return the three lexicographically minimums products. Return a list of lists of the suggested products after each character of searchWord is typed. Example 1: Input: products = ["mobile","mouse","moneypot","monitor","mousepad"], searchWord = "mouse" Output: [["mobile","moneypot","monitor"],["mobile","moneypot","monitor"],["mouse","mousepad"],["mouse" Explanation: products sorted lexicographically = ["mobile","moneypot","monitor","mouse","mousepad"]. After typing m and mo all products match and we show user ["mobile","moneypot","monitor"]. After typing mou, mous and mouse the system suggests ["mouse","mousepad"].

Example 2: Input: products = ["havana"], searchWord = "havana" Output: [["havana"],["havana"],["havana"],["havana"],["havana"],["havana"]] Explanation: The only word ”havana” will be always suggested while typing the search word.

Constraints:

1 <= products.length <= 1000 1 <= products[i].length <= 3000 1 <= sum(products[i].length) <= 2 * 104 All the strings of products are unique. products[i] consists of lowercase English letters. 1 <= searchWord.length <= 1000 searchWord consists of lowercase English letters.

==
Problem Number: 1078 URL: https://leetcode.com/problems/number-of-burgers-with-no-waste-of-ingredients Title: 1276. Number of Burgers with No Waste of Ingredients Problem Description: Given two integers tomatoSlices and cheeseSlices. The ingredients of different burgers are as follows:

Jumbo Burger: 4 tomato slices and 1 cheese slice. Small Burger: 2 Tomato slices and 1 cheese slice.

Return [total_jumbo, total_small] so that the number of remaining tomatoSlices equal to 0 and the number of remaining cheeseSlices equal to 0. If it is not possible to make the remaining tomatoSlices and cheeseSlices equal to 0 return []. Example 1: Input: tomatoSlices = 16, cheeseSlices = 7 Output: [1,6] Explantion: To make one jumbo burger and 6 small burgers we need 4*1 + 2*6 = 16 tomato and 1 + 6 = 7 cheese. There will be no remaining ingredients.

Example 2: Input: tomatoSlices = 17, cheeseSlices = 4 Output: [] Explantion: There will be no way to use all ingredients to make small and jumbo burgers.

Example 3: Input: tomatoSlices = 4, cheeseSlices = 17 Output: [] Explantion: Making 1 jumbo burger there will be 16 cheese remaining and making 2 small burgers there will be 15 cheese remaining.

Constraints:

0 <= tomatoSlices, cheeseSlices <= 107

==
Problem Number: 1079 URL: https://leetcode.com/problems/count-square-submatrices-with-all-ones Title: 1277. Count Square Submatrices with All Ones Problem Description: Given a m * n matrix of ones and zeros, return how many square submatrices have all ones. Example 1: Input: matrix = [[0,1,1,1], [1,1,1,1], [0,1,1,1]] Output: 15 Explanation: There are 10 squares of side 1. There are 4 squares of side 2. There is 1 square of side 3. Total number of squares = 10 + 4 + 1 = 15.

Example 2: Input: matrix = [[1,0,1], [1,1,0], [1,1,0]] Output: 7 Explanation: There are 6 squares of side 1. There is 1 square of side 2. Total number of squares = 6 + 1 = 7.

Constraints:

1 <= arr.length <= 300 1 <= arr[0].length <= 300 0 <= arr[i][j] <= 1

===
Problem Number: 1080 URL: https://leetcode.com/problems/group-the-people-given-the-group-size-they-belong-to Title: 1282. Group the People Given the Group Size They Belong To Problem Description: There are n people that are split into some unknown number of groups. Each person is labeled with a unique ID from 0 to n - 1. You are given an integer array groupSizes, where groupSizes[i] is the size of the group that person i is in. For example, if groupSizes[1] = 3, then person 1 must be in a group of size 3. Return a list of groups such that each person i is in a group of size groupSizes[i]. Each person should appear in exactly one group, and every person must be in a group. If there are multiple answers, return any of them. It is guaranteed that there will be at least one valid solution for the given input. Example 1: Input: groupSizes = [3,3,3,3,3,1,3] Output: [[5],[0,1,2],[3,4,6]] Explanation: The first group is [5]. The size is 1, and groupSizes[5] = 1. The second group is [0,1,2]. The size is 3, and groupSizes[0] = groupSizes[1] = groupSizes[2] = 3. The third group is [3,4,6]. The size is 3, and groupSizes[3] = groupSizes[4] = groupSizes[6] = 3. Other possible solutions are [[2,1,6],[5],[0,4,3]] and [[5],[0,6,2],[4,3,1]].

Example 2: Input: groupSizes = [2,1,3,3,3,2] Output: [[1],[0,5],[2,3,4]]

Constraints:

groupSizes.length == n 1 <= n <= 500 1 <= groupSizes[i] <= n

===
Problem Number: 1081 URL: https://leetcode.com/problems/find-the-smallest-divisor-given-a-threshold Title: 1283. Find the Smallest Divisor Given a Threshold Problem Description: Given an array of integers nums and an integer threshold, we will choose a positive integer divisor, divide all the array by it, and sum the division's result. Find the smallest divisor such that the result mentioned above is less than or equal to threshold. Each result of the division is rounded to the nearest integer greater than or equal to that element. (For example: 7/3 = 3 and 10/2 = 5). The test cases are generated so that there will be an answer. Example 1: Input: nums = [1,2,5,9], threshold = 6 Output: 5 Explanation: We can get a sum to 17 (1+2+5+9) if the divisor is 1. If the divisor is 4 we can get a sum of 7 (1+1+2+3) and if the divisor is 5 the sum will be 5 (1+1+1+2).

Example 2: Input: nums = [44,22,33,11,1], threshold = 5 Output: 44

Constraints:

1 <= nums.length <= 5 * 104 1 <= nums[i] <= 106 nums.length <= threshold <= 106

===
Problem Number: 1082 URL: https://leetcode.com/problems/iterator-for-combination Title: 1286. Iterator for Combination Problem Description:

Design the CombinationIterator class:

CombinationIterator(string characters, int combinationLength) Initializes the object with a string characters of sorted distinct lowercase English letters and a number combinationLength as arguments. next() Returns the next combination of length combinationLength in lexicographical order. hasNext() Returns true if and only if there exists a next combination.

Example 1: Input ["CombinationIterator", "next", "hasNext", "next", "hasNext", "next", "hasNext"] [["abc", 2], [], [], [], [], [], []] Output [null, "ab", true, "ac", true, "bc", false]

Explanation CombinationIterator itr = new CombinationIterator("abc", 2); itr.next(); // return "ab" itr.hasNext(); // return True itr.next(); // return "ac" itr.hasNext(); // return True itr.next(); // return "bc" itr.hasNext(); // return False

Constraints:

1 <= combinationLength <= characters.length <= 15 All the characters of characters are unique. At most 104 calls will be made to next and hasNext. It is guaranteed that all calls of the function next are valid.

==
Problem Number: 1083 URL: https://leetcode.com/problems/remove-covered-intervals Title: 1288. Remove Covered Intervals Problem Description: Given an array intervals where intervals[i] = [li, ri] represent the interval [li, ri), remove all intervals that are covered by another interval in the list. The interval [a, b) is covered by the interval [c, d) if and only if c <= a and b <= d. Return the number of remaining intervals. Example 1: Input: intervals = [[1,4],[3,6],[2,8]] Output: 2 Explanation: Interval [3,6] is covered by [2,8], therefore it is removed.

Example 2: Input: intervals = [[1,4],[2,3]] Output: 1

Constraints:

1 <= intervals.length <= 1000 intervals[i].length == 2 0 <= li < ri <= 105 All the given intervals are unique.

==
Problem Number: 1084 URL: https://leetcode.com/problems/sequential-digits Title: 1291. Sequential Digits Problem Description: An integer has sequential digits if and only if each digit in the number is one more than the previous digit. Return a sorted list of all the integers in the range [low, high] inclusive that have sequential digits. Example 1: Input: low = 100, high = 300 Output: [123,234] Example 2: Input: low = 1000, high = 13000 Output: [1234,2345,3456,4567,5678,6789,12345]

Constraints:

10 <= low <= high <= 10^9

==

Problem Number: 1085 URL: https://leetcode.com/problems/maximum-side-length-of-a-square-with-sum-less-than-or-equal-to-threshold Title: 1292. Maximum Side Length of a Square with Sum Less than or Equal to Threshold Problem Description: Given a m x n matrix mat and an integer threshold, return the maximum side-length of a square with a sum less than or equal to threshold or return 0 if there is no such square. Example 1:

Input: mat = [[1,1,3,2,4,3,2],[1,1,3,2,4,3,2],[1,1,3,2,4,3,2]], threshold = 4 Output: 2 Explanation: The maximum side length of square with sum less than 4 is 2 as shown.

Example 2: Input: mat = [[2,2,2,2,2],[2,2,2,2,2],[2,2,2,2,2],[2,2,2,2,2],[2,2,2,2,2]], threshold = 1 Output: 0

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 300 0 <= mat[i][j] <= 104 0 <= threshold <= 105

==

Problem Number: 1086 URL: https://leetcode.com/problems/divide-array-in-sets-of-k-consecutive-numbers Title: 1296. Divide Array in Sets of K Consecutive Numbers Problem Description: Given an array of integers nums and a positive integer k, check whether it is possible to divide this array into sets of k consecutive numbers. Return true if it is possible. Otherwise, return false. Example 1: Input: nums = [1,2,3,3,4,4,5,6], k = 4 Output: true Explanation: Array can be divided into [1,2,3,4] and [3,4,5,6].

Example 2: Input: nums = [3,2,1,2,3,4,3,4,5,9,10,11], k = 3 Output: true Explanation: Array can be divided into [1,2,3] , [2,3,4] , [3,4,5] and [9,10,11].

Example 3: Input: nums = [1,2,3,4], k = 3 Output: false Explanation: Each array should be divided in subarrays of size 3.

Constraints:

1 <= k <= nums.length <= 105 1 <= nums[i] <= 109

Note: This question is the same as 846: https://leetcode.com/problems/hand-of-straights/ ==
Problem Number: 1087 URL: https://leetcode.com/problems/maximum-number-of-occurrences-of-a-substring Title: 1297. Maximum Number of Occurrences of a Substring Problem Description: Given a string s, return the maximum number of occurrences of any substring under the following rules:

The number of unique characters in the substring must be less than or equal to maxLetters. The substring size must be between minSize and maxSize inclusive.

Example 1: Input: s = ”aababcaab”, maxLetters = 2, minSize = 3, maxSize = 4 Output: 2 Explanation: Substring ”aab” has 2 occurrences in the original string. It satisfies the conditions, 2 unique letters and size 3 (between minSize and maxSize).

Example 2: Input: s = "aaaa", maxLetters = 1, minSize = 3, maxSize = 3 Output: 2 Explanation: Substring "aaa" occur 2 times in the string. It can overlap.

Constraints:

1 <= s.length <= 105 1 <= maxLetters <= 26 1 <= minSize <= maxSize <= min(26, s.length) s consists of only lowercase English letters.

==
Problem Number: 1088 URL: https://leetcode.com/problems/sum-of-mutated-array-closest-to-target Title: 1300. Sum of Mutated Array Closest to Target Problem Description: Given an integer array arr and a target value target, return the integer value such that when we change all the integers larger than value in the given array to be equal to value, the sum of the array gets as close as possible (in absolute difference) to target. In case of a tie, return the minimum such integer. Notice that the answer is not neccesarilly a number from arr. Example 1: Input: arr = [4,9,3], target = 10 Output: 3 Explanation: When using 3 arr converts to [3, 3, 3] which sums 9 and that's the optimal answer.

Example 2: Input: arr = [2,3,5], target = 10 Output: 5

Example 3: Input: arr = [60864,25176,27249,21296,20204], target = 56803 Output: 11361

Constraints:

1 <= arr.length <= 104 1 <= arr[i], target <= 105

==
Problem Number: 1089 URL: https://leetcode.com/problems/deepest-leaves-sum Title: 1302. Deepest Leaves Sum Problem Description: Given the root of a binary tree, return the sum of values of its deepest leaves. Example 1:

Input: root = [1,2,3,4,5,null,6,7,null,null,null,null,8] Output: 15

Example 2: Input: root = [6,7,8,2,7,1,3,9,null,1,4,null,null,null,5] Output: 19

Constraints:

The number of nodes in the tree is in the range [1, 104]. 1 <= Node.val <= 100

==
Problem Number: 1090 URL: https://leetcode.com/problems/all-elements-in-two-binary-search-trees Title: 1305. All Elements in Two Binary Search Trees Problem Description: Given two binary search trees root1 and root2, return a list containing all the integers from both trees sorted in ascending order. Example 1:

Input: root1 = [2,1,4], root2 = [1,0,3] Output: [0,1,1,2,3,4]

Example 2:

Input: root1 = [1,null,8], root2 = [8,1] Output: [1,1,8,8]

Constraints:

The number of nodes in each tree is in the range [0, 5000]. -105 <= Node.val <= 105

==
Problem Number: 1091 URL: https://leetcode.com/problems/jump-game-iii Title: 1306. Jump Game III Problem Description: Given an array of non-negative integers arr, you are initially positioned at start index of the array. When you are at index i, you can jump to i + arr[i] or i - arr[i], check if you can reach any index with value 0. Notice that you can not jump outside of the array at any time. Example 1: Input: arr = [4,2,3,0,3,1,2], start = 5 Output: true Explanation: All possible ways to reach at index 3 with value 0 are: index 5 -> index 4 -> index 1 -> index 3 index 5 -> index 6 -> index 4 -> index 1 -> index 3

Example 2: Input: arr = [4,2,3,0,3,1,2], start = 0 Output: true Explanation: One possible way to reach at index 3 with value 0 is: index 0 -> index 4 -> index 1 -> index 3

Example 3: Input: arr = [3,0,2,1,2], start = 2 Output: false Explanation: There is no way to reach at index 1 with value 0.

Constraints:

1 <= arr.length <= 5 * 104 0 <= arr[i] < arr.length 0 <= start < arr.length

==
Problem Number: 1092 URL: https://leetcode.com/problems/xor-queries-of-a-subarray Title: 1310. XOR Queries of a Subarray Problem Description: You are given an array arr of positive integers. You are also given the array queries where queries[i] = [lefti, righti]. For each query i compute the XOR of elements from lefti to righti (that is, arr[lefti] XOR arr[lefti + 1] XOR ... XOR arr[righti]). Return an array answer where answer[i] is the answer to the ith query. Example 1: Input: arr = [1,3,4,8], queries = [[0,1],[1,2],[0,3],[3,3]] Output: [2,7,14,8] Explanation: The binary representation of the elements in the array are: 1 = 0001 3 = 0011 4 = 0100 8 = 1000 The XOR values for queries are: [0,1] = 1 xor 3 = 2 [1,2] = 3 xor 4 = 7 [0,3] = 1 xor 3 xor 4 xor 8 = 14 [3,3] = 8

Example 2: Input: arr = [4,8,2,10], queries = [[2,3],[1,3],[0,0],[0,3]] Output: [8,0,4,4]

Constraints:

1 <= arr.length, queries.length <= 3 * 104 1 <= arr[i] <= 109 queries[i].length == 2 0 <= lefti <= righti < arr.length

==
Problem Number: 1093 URL: https://leetcode.com/problems/get-watched-videos-by-your-friends Title: 1311. Get Watched Videos by Your Friends Problem Description: There are n people, each person has a unique id between 0 and n-1. Given the arrays watchedVideos and friends, where watchedVideos[i] and friends[i] contain the list of watched videos and the list of friends respectively for the person with id = i. Level 1 of videos are all watched videos by your friends, level 2 of videos are all watched videos by the friends of your friends and so on. In general, the level k of videos are all watched videos by people with the shortest path exactly equal to k with you. Given your id and the level of videos, return the list of videos ordered by their frequencies (increasing). For videos with the same frequency order them alphabetically from least to greatest. Example 1:

Input: watchedVideos = [["A","B"],["C"],["B","C"],["D"]], friends = [[1,2],[0,3],[0,3],[1,2]], id = 0, level = 1 Output: ["B","C"] Explanation: You have id = 0 (green color in the figure) and your friends are (yellow color in the figure): Person with id = 1 -> watchedVideos = ["C"] Person with id = 2 -> watchedVideos = ["B","C"] The frequencies of watchedVideos by your friends are: B -> 1 C -> 2

Example 2:

Input: watchedVideos = [["A","B"],["C"],["B","C"],["D"]], friends = [[1,2],[0,3],[0,3],[1,2]], id = 0, level = 2 Output: ["D"] Explanation: You have id = 0 (green color in the figure) and the only friend of your friends is the person with id = 3 (yellow color in the figure).

Constraints:

n == watchedVideos.length == friends.length 2 <= n <= 100 1 <= watchedVideos[i].length <= 100 1 <= watchedVideos[i][j].length <= 8 0 <= friends[i].length < n 0 <= friends[i][j] < n 0 <= id < n 1 <= level < n if friends[i] contains j, then friends[j] contains i

==
Problem Number: 1094 URL: https://leetcode.com/problems/matrix-block-sum Title: 1314. Matrix Block Sum Problem Description: Given a m x n matrix mat and an integer k, return a matrix answer where each answer[i][j] is the sum of all elements mat[r][c] for:

i - k <= r <= i + k, j - k <= c <= j + k, and (r, c) is a valid position in the matrix.

Example 1: Input: mat = [[1,2,3],[4,5,6],[7,8,9]], k = 1 Output: [[12,21,16],[27,45,33],[24,39,28]]

Example 2: Input: mat = [[1,2,3],[4,5,6],[7,8,9]], k = 2 Output: [[45,45,45],[45,45,45],[45,45,45]]

Constraints:

m == mat.length n == mat[i].length 1 <= m, n, k <= 100 1 <= mat[i][j] <= 100

==
Problem Number: 1095 URL: https://leetcode.com/problems/sum-of-nodes-with-even-valued-grandparent Title: 1315. Sum of Nodes with Even-Valued Grandparent Problem Description: Given the root of a binary tree, return the sum of values of nodes with an even-valued grandparent. If there are no nodes with an even-valued grandparent, return 0. A grandparent of a node is the parent of its parent if it exists. Example 1:

Input: root = [6,7,8,2,7,1,3,9,null,1,4,null,null,null,5] Output: 18 Explanation: The red nodes are the nodes with even-value grandparent while the blue nodes are the even-value grandparents.

Example 2:

Input: root = [1] Output: 0

Constraints:

The number of nodes in the tree is in the range [1, 104]. 1 <= Node.val <= 100

==
Problem Number: 1096 URL: https://leetcode.com/problems/minimum-flips-to-make-a-or-b-equal-to-c Title: 1318. Minimum Flips to Make a OR b Equal to c Problem Description: Given 3 positives numbers a, b and c. Return the minimum flips required in some bits of a and b to make (a OR b == c). (bitwise OR operation). Flip operation consists of change any single bit 1 to 0 or change the bit 0 to 1 in their binary representation. Example 1:

Input: a = 2, b = 6, c = 5 Output: 3 Explanation: After flips a = 1 , b = 4 , c = 5 such that (a OR b == c) Example 2: Input: a = 4, b = 2, c = 7 Output: 1

Example 3: Input: a = 1, b = 2, c = 3 Output: 0

Constraints:

1 <= a <= 10^9 1 <= b <= 10^9 1 <= c <= 10^9

==
Problem Number: 1097 URL: https://leetcode.com/problems/number-of-operations-to-make-network-connected Title: 1319. Number of Operations to Make Network Connected Problem Description: There are n computers numbered from 0 to n - 1 connected by ethernet cables connections forming a network where connections[i] = [ai, bi] represents a connection between computers ai and bi. Any computer can reach any other computer directly or indirectly through the network. You are given an initial computer network connections. You can extract certain cables between two directly connected

computers, and place them between any pair of disconnected computers to make them directly connected. Return the minimum number of times you need to do this in order to make all the computers connected. If it is not possible, return -1. Example 1:

Input: n = 4, connections = [[0,1],[0,2],[1,2]] Output: 1 Explanation: Remove cable between computer 1 and 2 and place between computers 1 and 3.

Example 2:

Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2],[1,3]] Output: 2

Example 3: Input: n = 6, connections = [[0,1],[0,2],[0,3],[1,2]] Output: -1 Explanation: There are not enough cables.

Constraints:

1 <= n <= 105 1 <= connections.length <= min(n * (n - 1) / 2, 105) connections[i].length == 2 0 <= ai, bi < n ai != bi There are no repeated connections. No two computers are connected by more than one cable.

==
Problem Number: 1098 URL: https://leetcode.com/problems/print-words-vertically Title: 1324. Print Words Vertically Problem Description: Given a string s. Return all the words vertically in the same order in which they appear in s. Words are returned as a list of strings, complete with spaces when is necessary. (Trailing spaces are not allowed). Each word would be put on only one column and that in one column there will be only one word. Example 1: Input: s = "HOW ARE YOU" Output: ["HAY","ORO","WEU"] Explanation: Each word is printed vertically. "HAY" "ORO" "WEU"

Example 2: Input: s = "TO BE OR NOT TO BE" Output: ["TBONTB","OEROOE"," T"] Explanation: Trailing spaces is not allowed. "TBONTB" "OEROOE" " T"

Example 3: Input: s = "CONTEST IS COMING" Output: ["CIC","OSO","N M","T I","E N","S G","T"]

Constraints:

1 <= s.length <= 200 s contains only upper case English letters. It's guaranteed that there is only one space between 2 words.

==
Problem Number: 1099 URL: https://leetcode.com/problems/delete-leaves-with-a-given-value Title: 1325. Delete Leaves With a Given Value Problem Description: Given a binary tree root and an integer target, delete all the leaf nodes with value target. Note that once you delete a leaf node with value target, if its parent node becomes a leaf node and has the value target, it should also be deleted (you need to continue doing that until you cannot). Example 1:

Input: root = [1,2,3,2,null,2,4], target = 2 Output: [1,null,3,null,4] Explanation: Leaf nodes in green with value (target = 2) are removed (Picture in left). After removing, new nodes become leaf nodes with value (target = 2) (Picture in center).

Example 2:

Input: root = [1,3,3,3,2], target = 3 Output: [1,3,null,null,2]

Example 3:

Input: root = [1,2,null,2,null,2], target = 2 Output: [1] Explanation: Leaf nodes in green with value (target = 2) are removed at each step.

Constraints:

The number of nodes in the tree is in the range [1, 3000]. 1 <= Node.val, target <= 1000

==
Problem Number: 1100 URL: https://leetcode.com/problems/break-a-palindrome Title: 1328. Break a Palindrome Problem Description: Given a palindromic string of lowercase English letters palindrome, replace exactly one character with any lowercase English letter so that the resulting string is not a palindrome and that it is the lexicographically smallest one possible. Return the resulting string. If there is no way to replace a character to make it not a palindrome, return an empty string. A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, a has a character strictly smaller than the corresponding character in b. For example, "abcc" is lexicographically smaller than "abcd" because the first position they differ is at the fourth character, and 'c' is smaller than 'd'. Example 1: Input: palindrome = "abccba" Output: "aaccba" Explanation: There are many ways to make "abccba" not a palindrome, such as "zbccba", "aaccba", and "abacba". Of all the ways, "aaccba" is the lexicographically smallest.

Example 2: Input: palindrome = "a" Output: "" Explanation: There is no way to replace a single character to make "a" not a palindrome, so return an empty string.

Constraints:

1 <= palindrome.length <= 1000 palindrome consists of only lowercase English letters.

==
Problem Number: 1101 URL: https://leetcode.com/problems/sort-the-matrix-diagonally Title: 1329. Sort the Matrix Diagonally Problem Description: A matrix diagonal is a diagonal line of cells starting from some cell in either the topmost row or leftmost column and going in the bottom-right direction until reaching the matrix's end. For example, the matrix diagonal starting from mat[2][0], where mat is a 6 x 3 matrix, includes cells mat[2][0], mat[3][1], and

mat[4][2]. Given an m x n matrix mat of integers, sort each matrix diagonal in ascending order and return the resulting matrix. Example 1:

Input: mat = [[3,3,1,1],[2,2,1,2],[1,1,1,2]] Output: [[1,1,1,1],[1,2,2,2],[1,2,3,3]]

Example 2: Input: mat = [[11,25,66,1,69,7],[23,55,17,45,15,52],[75,31,36,44,58,8],[22,27,33,25,68,4],[84,28,14,11,5
Output: [[5,17,4,1,52,7],[11,11,25,45,8,69],[14,23,25,44,58,15],[22,27,31,36,50,66],[84,28,75,33,55,68]]

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 100 1 <= mat[i][j] <= 100

==
Problem Number: 1102 URL: https://leetcode.com/problems/filter-restaurants-by-vegan-friendly-price-and-distance Title: 1333. Filter Restaurants by Vegan-Friendly, Price and Distance Problem Description: Given the array restaurants where restaurants[i] = [idi, ratingi, veganFriendlyi, pricei, distancei]. You have to filter the restaurants using three filters. The veganFriendly filter will be either true (meaning you should only include restaurants with veganFriendlyi set to true) or false (meaning you can include any restaurant). In addition, you have the filters maxPrice and maxDistance which are the maximum value for price and distance of restaurants you should consider respectively. Return the array of restaurant IDs after filtering, ordered by rating from highest to lowest. For restaurants with the same rating, order them by id from highest to lowest. For simplicity veganFriendlyi and veganFriendly take value 1 when it is true, and 0 when it is false. Example 1: Input: restaurants = [[1,4,1,40,10],[2,8,0,50,5],[3,8,1,30,4],[4,10,0,10,3],[5,1,1,15,1]], veganFriendly = 1, maxPrice = 50, maxDistance = 10 Output: [3,1,5] Explanation: The restaurants are: Restaurant 1 [id=1, rating=4, veganFriendly=1, price=40, distance=10] Restaurant 2 [id=2, rating=8, veganFriendly=0, price=50, distance=5] Restaurant 3 [id=3, rating=8, veganFriendly=1, price=30, distance=4] Restaurant 4 [id=4, rating=10, veganFriendly=0, price=10, distance=3] Restaurant 5 [id=5, rating=1, veganFriendly=1, price=15, distance=1] After filter restaurants with veganFriendly = 1, maxPrice = 50 and maxDistance = 10 we have restaurant 3, restaurant 1 and restaurant 5 (ordered by rating from highest to lowest).

Example 2: Input: restaurants = [[1,4,1,40,10],[2,8,0,50,5],[3,8,1,30,4],[4,10,0,10,3],[5,1,1,15,1]], veganFriendly = 0, maxPrice = 50, maxDistance = 10 Output: [4,3,2,1,5] Explanation: The restaurants are the same as in example 1, but in this case the filter veganFriendly = 0, therefore all restaurants are considered.

Example 3: Input: restaurants = [[1,4,1,40,10],[2,8,0,50,5],[3,8,1,30,4],[4,10,0,10,3],[5,1,1,15,1]], veganFriendly = 0, maxPrice = 30, maxDistance = 3 Output: [4,5]

Constraints:

1 <= restaurants.length <= 10^4 restaurants[i].length == 5 1 <= idi, ratingi, pricei, distancei <= 10^5 1 <= maxPrice, maxDistance <= 10^5 veganFriendlyi

and veganFriendly are 0 or 1. All idi are distinct.

== Problem Number: 1103 URL: https://leetcode.com/problems/find-the-city-with-the-smallest-number-of-neighbors-at-a-threshold-distance Title: 1334. Find the City With the Smallest Number of Neighbors at a Threshold Distance Problem Description: There are n cities numbered from 0 to n-1. Given the array edges where edges[i] = [fromi, toi, weighti] represents a bidirectional and weighted edge between cities fromi and toi, and given the integer distanceThreshold. Return the city with the smallest number of cities that are reachable through some path and whose distance is at most distanceThreshold, If there are multiple such cities, return the city with the greatest number. Notice that the distance of a path connecting cities i and j is equal to the sum of the edges' weights along that path. Example 1:

Input: n = 4, edges = [[0,1,3],[1,2,1],[1,3,4],[2,3,1]], distanceThreshold = 4 Output: 3 Explanation: The figure above describes the graph. The neighboring cities at a distanceThreshold = 4 for each city are: City 0 -> [City 1, City 2] City 1 -> [City 0, City 2, City 3] City 2 -> [City 0, City 1, City 3] City 3 -> [City 1, City 2] Cities 0 and 3 have 2 neighboring cities at a distanceThreshold = 4, but we have to return city 3 since it has the greatest number.

Example 2:

Input: n = 5, edges = [[0,1,2],[0,4,8],[1,2,3],[1,4,2],[2,3,1],[3,4,1]], distanceThreshold = 2 Output: 0 Explanation: The figure above describes the graph. The neighboring cities at a distanceThreshold = 2 for each city are: City 0 -> [City 1] City 1 -> [City 0, City 4] City 2 -> [City 3, City 4] City 3 -> [City 2, City 4] City 4 -> [City 1, City 2, City 3] The city 0 has 1 neighboring city at a distanceThreshold = 2.

Constraints:

2 <= n <= 100 1 <= edges.length <= n * (n - 1) / 2 edges[i].length == 3 0 <= fromi < toi < n 1 <= weighti, distanceThreshold <= 10^4 All pairs (fromi, toi) are distinct.

== Problem Number: 1104 URL: https://leetcode.com/problems/reduce-array-size-to-the-half Title: 1338. Reduce Array Size to The Half Problem Description: You are given an integer array arr. You can choose a set of integers and remove all the occurrences of these integers in the array. Return the minimum size of the set so that at least half of the integers of the array are removed. Example 1: Input: arr = [3,3,3,3,5,5,5,2,2,7] Output: 2 Explanation: Choosing {3,7} will make the new array [5,5,5,2,2] which has size 5 (i.e equal to half of the size of the old array). Possible sets of size 2 are {3,5},{3,2},{5,2}. Choosing set {2,7} is not possible as it will make the new array [3,3,3,3,5,5,5] which has a size greater than half of the size of the old array.

Example 2: Input: arr = [7,7,7,7,7,7] Output: 1 Explanation: The only possible set you can choose is {7}. This will make the new array empty.

Constraints:

2 <= arr.length <= 105 arr.length is even. 1 <= arr[i] <= 105

==

Problem Number: 1105 URL: https://leetcode.com/problems/maximum-product-of-splitted-binary-tree Title: 1339. Maximum Product of Splitted Binary Tree Problem Description: Given the root of a binary tree, split the binary tree into two subtrees by removing one edge such that the product of the sums of the subtrees is maximized. Return the maximum product of the sums of the two subtrees. Since the answer may be too large, return it modulo 109 + 7. Note that you need to maximize the answer before taking the mod and not after taking it. Example 1:

Input: root = [1,2,3,4,5,6] Output: 110 Explanation: Remove the red edge and get 2 binary trees with sum 11 and 10. Their product is 110 (11*10)

Example 2:

Input: root = [1,null,2,3,4,null,null,5,6] Output: 90 Explanation: Remove the red edge and get 2 binary trees with sum 15 and 6.Their product is 90 (15*6)

Constraints:

The number of nodes in the tree is in the range [2, 5 * 104]. 1 <= Node.val <= 104

==

Problem Number: 1106 URL: https://leetcode.com/problems/number-of-sub-arrays-of-size-k-and-average-greater-than-or-equal-to-threshold Title: 1343. Number of Sub-arrays of Size K and Average Greater than or Equal to Threshold Problem Description: Given an array of integers arr and two integers k and threshold, return the number of sub-arrays of size k and average greater than or equal to threshold. Example 1: Input: arr = [2,2,2,2,5,5,5,8], k = 3, threshold = 4 Output: 3 Explanation: Sub-arrays [2,5,5],[5,5,5] and [5,5,8] have averages 4, 5 and 6 respectively. All other sub-arrays of size 3 have averages less than 4 (the threshold).

Example 2: Input: arr = [11,13,17,23,29,31,7,5,2,3], k = 3, threshold = 5 Output: 6 Explanation: The first 6 sub-arrays of size 3 have averages greater than 5. Note that averages are not integers.

Constraints:

1 <= arr.length <= 105 1 <= arr[i] <= 104 1 <= k <= arr.length 0 <= threshold <= 104

==

Problem Number: 1107 URL: https://leetcode.com/problems/angle-between-hands-of-a-clock Title: 1344. Angle Between Hands of a Clock Problem Description: Given two numbers, hour and minutes, return the smaller angle (in degrees) formed between the hour and the minute hand. Answers within 10-5 of the actual value will be accepted as correct. Example 1:

Input: hour = 12, minutes = 30 Output: 165

Example 2:

Input: hour = 3, minutes = 30 Output: 75

Example 3:

Input: hour = 3, minutes = 15 Output: 7.5

Constraints:

1 <= hour <= 12 0 <= minutes <= 59

== Problem Number: 1108 URL: https://leetcode.com/problems/minimum-number-of-steps-to-make-two-strings-anagram Title: 1347. Minimum Number of Steps to Make Two Strings Anagram Problem Description: You are given two strings of the same length s and t. In one step you can choose any character of t and replace it with another character. Return the minimum number of steps to make t an anagram of s. An Anagram of a string is a string that contains the same characters with a different (or the same) ordering. Example 1: Input: s = "bab", t = "aba" Output: 1 Explanation: Replace the first 'a' in t with b, t = "bba" which is anagram of s.

Example 2: Input: s = "leetcode", t = "practice" Output: 5 Explanation: Replace 'p', 'r', 'a', 'i' and 'c' from t with proper characters to make t anagram of s.

Example 3: Input: s = "anagram", t = "mangaar" Output: 0 Explanation: "anagram" and "mangaar" are anagrams.

Constraints:

1 <= s.length <= 5 * 104 s.length == t.length s and t consist of lowercase English letters only.

== Problem Number: 1109 URL: https://leetcode.com/problems/tweet-counts-per-frequency Title: 1348. Tweet Counts Per Frequency Problem Description: A social media company is trying to monitor activity on their site by analyzing the number of tweets that occur in select periods of time. These periods can be partitioned into smaller time chunks based on a certain frequency (every minute, hour, or day). For example, the period [10, 10000] (in seconds) would be partitioned into the following time chunks with these frequencies:

Every minute (60-second chunks): [10,69], [70,129], [130,189], ..., [9970,10000] Every hour (3600-second chunks): [10,3609], [3610,7209], [7210,10000] Every day (86400-second chunks): [10,10000]

Notice that the last chunk may be shorter than the specified frequency's chunk size and will always end with the end time of the period (10000 in the above example). Design and implement an API to help the company with their analysis. Implement the TweetCounts class:

TweetCounts() Initializes the TweetCounts object. void recordTweet(String tweetName, int time) Stores the tweetName at the recorded time (in seconds). List<Integer> getTweetCountsPerFrequency(String freq, String tweetName, int startTime, int endTime) Returns a list of integers representing the number of tweets with tweetName in each time chunk for the given period of time [startTime, endTime] (in seconds) and frequency freq.

freq is one of "minute", "hour", or "day" representing a frequency of every minute, hour, or day respectively.

Example: Input ["TweetCounts","recordTweet","recordTweet","recordTweet","getTweetCountsPerFrequency"
[[],["tweet3",0],["tweet3",60],["tweet3",10],["minute","tweet3",0,59],["minute","tweet3",0,60],["tweet3",120],["ho

Output [null,null,null,null,[2],[2,1],null,[4]]

Explanation TweetCounts tweetCounts = new TweetCounts(); tweetCounts.recordTweet("tweet3", 0); // New tweet "tweet3" at time 0 tweetCounts.recordTweet("tweet3", 60); // New tweet "tweet3" at time 60 tweetCounts.recordTweet("tweet3", 10); // New tweet "tweet3" at time 10 tweetCounts.getTweetCountsPerFrequency("minute", "tweet3", 0, 59); // return [2]; chunk [0,59] had 2 tweets tweetCounts.getTweetCountsPerFrequency("minute", "tweet3", 0, 60); // return [2,1]; chunk [0,59] had 2 tweets, chunk [60,60] had 1 tweet tweetCounts.recordTweet("tweet3", 120); // New tweet "tweet3" at time 120 tweetCounts.getTweetCountsPerFrequency("hour", "tweet3", 0, 210); // return [4]; chunk [0,210] had 4 tweets

Constraints:

0 <= time, startTime, endTime <= 109 0 <= endTime - startTime <= 104 There will be at most 104 calls in total to recordTweet and getTweetCountsPerFrequency.

==
Problem Number: 1110 URL: https://leetcode.com/problems/product-of-the-last-k-numbers Title: 1352. Product of the Last K Numbers Problem Description: Design an algorithm that accepts a stream of integers and retrieves the product of the last k integers of the stream. Implement the ProductOfNumbers class:

ProductOfNumbers() Initializes the object with an empty stream. void add(int num) Appends the integer num to the stream. int getProduct(int k) Returns the product of the last k numbers in the current list. You can assume that always the current list has at least k numbers.

The test cases are generated so that, at any time, the product of any contiguous sequence of numbers will fit into a single 32-bit integer without overflowing. Example: Input ["ProductOfNumbers","add","add","add","add","add","getProduct","getProduct","getProduct" [[],[3],[0],[2],[5],[4],[2],[3],[4],[8],[2]]

Output [null,null,null,null,null,null,20,40,0,null,32]

Explanation ProductOfNumbers productOfNumbers = new ProductOfNumbers(); productOfNumbers.add(3); // [3] productOfNumbers.add(0); // [3,0] productOfNumbers.add(2); // [3,0,2] productOfNumbers.add(5); // [3,0,2,5] productOfNumbers.add(4); // [3,0,2,5,4] productOfNumbers.getProduct(2); // return 20. The product of the last 2 numbers is 5 * 4 = 20 productOfNumbers.getProduct(3); // return 40. The product of the last 3 numbers is 2 * 5 * 4 = 40 productOfNumbers.getProduct(4); // return 0. The product of the last 4 numbers is 0 * 2 * 5 * 4 = 0 productOfNumbers.add(8); // [3,0,2,5,4,8] productOfNumbers.getProduct(2); // return 32. The product of the last 2 numbers is 4 * 8 = 32

Constraints:

0 <= num <= 100 1 <= k <= 4 * 104 At most 4 * 104 calls will be made to add and getProduct. The product of the stream at any point in time will fit in a 32-bit integer.

==
Problem Number: 1111 URL: https://leetcode.com/problems/maximum-number-of-events-that-can-be-attended Title: 1353. Maximum Number of Events That Can Be Attended Problem Description: You are given an array of events where events[i] = [startDayi, endDayi]. Every event i starts at startDayi and ends at endDayi. You can attend an event i at any day d where startTimei <= d <= endTimei. You can only attend one event at any time d. Return the maximum number of events you can attend. Example 1:

Input: events = [[1,2],[2,3],[3,4]] Output: 3 Explanation: You can attend all the three events. One way to attend them all is as shown. Attend the first event on day 1. Attend the second event on day 2. Attend the third event on day 3.

Example 2: Input: events= [[1,2],[2,3],[3,4],[1,2]] Output: 4

Constraints:

1 <= events.length <= 105 events[i].length == 2 1 <= startDayi <= endDayi <= 105

==
Problem Number: 1112 URL: https://leetcode.com/problems/apply-discount-every-n-orders Title: 1357. Apply Discount Every n Orders Problem Description: There is a supermarket that is frequented by many customers. The products sold at the supermarket are represented as two parallel integer arrays products and prices, where the ith product has an ID of products[i] and a price of prices[i]. When a customer is paying, their bill is represented

as two parallel integer arrays product and amount, where the jth product they purchased has an ID of product[j], and amount[j] is how much of the product they bought. Their subtotal is calculated as the sum of each amount[j] * (price of the jth product). The supermarket decided to have a sale. Every nth customer paying for their groceries will be given a percentage discount. The discount amount is given by discount, where they will be given discount percent off their subtotal. More formally, if their subtotal is bill, then they would actually pay bill * ((100 - discount) / 100). Implement the Cashier class:

Cashier(int n, int discount, int[] products, int[] prices) Initializes the object with n, the discount, and the products and their prices. double getBill(int[] product, int[] amount) Returns the final total of the bill with the discount applied (if any). Answers within 10-5 of the actual value will be accepted.

Example 1: Input ["Cashier","getBill","getBill","getBill","getBill","getBill","getBill","getBill"] [[3,50,[1,2,3,4,5,6,7],[100,200,300,400,300,200,100]],[[1,2],[1,2]],[[3,7],[10,10]],[[1,2,3,4,5,6,7],[1,1,1,1,1,1,1]],[[4],[10 Output [null,500.0,4000.0,800.0,4000.0,4000.0,7350.0,2500.0] Explanation Cashier cashier = new Cashier(3,50,[1,2,3,4,5,6,7],[100,200,300,400,300,200,100]); cashier.getBill([1,2],[1,2]); // return 500.0. 1st customer, no discount. // bill = 1 * 100 + 2 * 200 = 500. cashier.getBill([3,7],[10,10]); // return 4000.0. 2nd customer, no discount. // bill = 10 * 300 + 10 * 100 = 4000. cashier.getBill([1,2,3,4,5,6,7],[1,1,1,1,1,1,1]); // return 800.0. 3rd customer, 50% discount. // Original bill = 1600 // Actual bill = 1600 * ((100 - 50) / 100) = 800. cashier.getBill([4],[10]); // return 4000.0. 4th customer, no discount. cashier.getBill([7,3],[10,10]); // return 4000.0. 5th customer, no discount. cashier.getBill([7,5,3,1,6,4,2],[10,10,10,9,9,9,7]); // return 7350.0. 6th customer, 50% discount. // Original bill = 14700, but with // Actual bill = 14700 * ((100 - 50) / 100) = 7350. cashier.getBill([2,3,5],[5,3,2]); // return 2500.0. 7th customer, no discount.

Constraints:

1 <= n <= 104 0 <= discount <= 100 1 <= products.length <= 200 prices.length == products.length 1 <= products[i] <= 200 1 <= prices[i] <= 1000 The elements in products are unique. 1 <= product.length <= products.length amount.length == product.length product[j] exists in products. 1 <= amount[j] <= 1000 The elements of product are unique. At most 1000 calls will be made to getBill. Answers within 10-5 of the actual value will be accepted.

==
Problem Number: 1113 URL: https://leetcode.com/problems/number-of-substrings-containing-all-three-characters Title: 1358. Number of Substrings Containing All Three Characters Problem Description: Given a string s consisting only of characters a, b and c. Return the number of substrings containing at least one occurrence of all these characters a, b and c. Example 1: Input: s = "abcabc" Output: 10 Explanation: The substrings containing at least one occurrence of the characters a, b and c are "abc", "abca", "abcab", "abcabc",

”bca”, ”bcab”, ”bcabc”, ”cab”, ”cabc” and ”abc” (again).

Example 2: Input: s = ”aaacb” Output: 3 Explanation: The substrings containing at least one occurrence of the characters a, b and c are ”aaacb”, ”aacb” and ”acb”.

Example 3: Input: s = ”abc” Output: 1

Constraints:

3 <= s.length <= 5 x 10^4 s only consists of a, b or c characters.

==
Problem Number: 1114 URL: https://leetcode.com/problems/validate-binary-tree-nodes Title: 1361. Validate Binary Tree Nodes Problem Description: You have n binary tree nodes numbered from 0 to n - 1 where node i has two children leftChild[i] and rightChild[i], return true if and only if all the given nodes form exactly one valid binary tree. If node i has no left child then leftChild[i] will equal -1, similarly for the right child. Note that the nodes have no values and that we only use the node numbers in this problem. Example 1:

Input: n = 4, leftChild = [1,-1,3,-1], rightChild = [2,-1,-1,-1] Output: true

Example 2:

Input: n = 4, leftChild = [1,-1,3,-1], rightChild = [2,3,-1,-1] Output: false

Example 3:

Input: n = 2, leftChild = [1,0], rightChild = [-1,-1] Output: false

Constraints:

n == leftChild.length == rightChild.length 1 <= n <= 104 -1 <= leftChild[i], rightChild[i] <= n - 1

==
Problem Number: 1115 URL: https://leetcode.com/problems/closest-divisors Title: 1362. Closest Divisors Problem Description: Given an integer num, find the closest two integers in absolute difference whose product equals num + 1 or num + 2. Return the two integers in any order. Example 1: Input: num = 8 Output: [3,3] Explanation: For num + 1 = 9, the closest divisors are 3 & 3, for num + 2 = 10, the closest divisors are 2 & 5, hence 3 & 3 is chosen.

Example 2: Input: num = 123 Output: [5,25]

Example 3: Input: num = 999 Output: [40,25]

Constraints:

1 <= num <= 10^9

==
Problem Number: 1116 URL: https://leetcode.com/problems/rank-teams-by-votes Title: 1366. Rank Teams by Votes Problem Description: In a special ranking system, each voter gives a rank from highest to lowest to all teams participating in the competition. The ordering of teams is decided by who received the most position-one votes. If two or more teams tie in the first position, we consider the second position to resolve the conflict, if they tie again, we continue this process until the ties are resolved. If two or more teams are still tied after considering all positions, we rank them alphabetically based on their team letter. You are given an array of strings votes which is the votes of all voters in the ranking systems. Sort all teams according to the ranking system described above. Return a string of all teams sorted by the ranking system. Example 1: Input: votes = ["ABC","ACB","ABC","ACB","ACB"] Output: "ACB" Explanation: Team A was ranked first place by 5 voters. No other team was voted as first place, so team A is the first team. Team B was ranked second by 2 voters and ranked third by 3 voters. Team C was ranked second by 3 voters and ranked third by 2 voters. As most of the voters ranked C second, team C is the second team, and team B is the third.

Example 2: Input: votes = ["WXYZ","XYZW"] Output: "XWYZ" Explanation: X is the winner due to the tie-breaking rule. X has the same votes as W for the first position, but X has one vote in the second position, while W does not have any votes in the second position.

Example 3: Input: votes = ["ZMNAGUEDSJYLBOPHRQICWFXTVK"] Output: "ZMNAGUEDSJYLBOPHRQICWFXTVK" Explanation: Only one voter, so their votes are used for the ranking.

Constraints:

1 <= votes.length <= 1000 1 <= votes[i].length <= 26 votes[i].length == votes[j].length for 0 <= i, j < votes.length. votes[i][j] is an English uppercase letter. All characters of votes[i] are unique. All the characters that occur in votes[0] also occur in votes[j] where 1 <= j < votes.length.

==
Problem Number: 1117 URL: https://leetcode.com/problems/linked-list-in-binary-tree Title: 1367. Linked List in Binary Tree Problem Description: Given a binary tree root and a linked list with head as the first node. Return True if all the elements in the linked list starting from the head correspond to some downward path connected in the binary tree otherwise return False. In this context downward path means a path that starts at some node and goes downwards. Example 1:

Input: head = [4,2,8], root = [1,4,4,null,2,2,null,1,null,6,8,null,null,null,null,1,3] Output: true Explanation: Nodes in blue form a subpath in the binary Tree.

Example 2:

Input: head = [1,4,2,6], root = [1,4,4,null,2,2,null,1,null,6,8,null,null,null,null,1,3]
Output: true

Example 3: Input: head = [1,4,2,6,8], root = [1,4,4,null,2,2,null,1,null,6,8,null,null,null,null,1,3]
Output: false Explanation: There is no path in the binary tree that contains all the elements of the linked list from head.

Constraints:

The number of nodes in the tree will be in the range [1, 2500]. The number of nodes in the list will be in the range [1, 100]. 1 <= Node.val <= 100 for each node in the linked list and binary tree.

==
Problem Number: 1118 URL: https://leetcode.com/problems/find-the-longest-substring-containing-vowels-in-even-counts Title: 1371. Find the Longest Substring Containing Vowels in Even Counts Problem Description: Given the string s, return the size of the longest substring containing each vowel an even number of times. That is, 'a', 'e', 'i', 'o', and 'u' must appear an even number of times. Example 1: Input: s = "eleetminicoworoep" Output: 13 Explanation: The longest substring is "leetminicowor" which contains two each of the vowels: e, i and o and zero of the vowels: a and u.

Example 2: Input: s = "leetcodeisgreat" Output: 5 Explanation: The longest substring is "leetc" which contains two e's.

Example 3: Input: s = "bcbcbc" Output: 6 Explanation: In this case, the given string "bcbcbc" is the longest because all vowels: a, e, i, o and u appear zero times.

Constraints:

1 <= s.length <= 5 x 10^5 s contains only lowercase English letters.

==
Problem Number: 1119 URL: https://leetcode.com/problems/longest-zigzag-path-in-a-binary-tree Title: 1372. Longest ZigZag Path in a Binary Tree Problem Description: You are given the root of a binary tree. A ZigZag path for a binary tree is defined as follow:

Choose any node in the binary tree and a direction (right or left). If the current direction is right, move to the right child of the current node; otherwise, move to the left child. Change the direction from right to left or from left to right. Repeat the second and third steps until you can't move in the tree.

Zigzag length is defined as the number of nodes visited - 1. (A single node has a length of 0). Return the longest ZigZag path contained in that tree. Example 1:

Input: root = [1,null,1,1,1,null,null,1,1,null,1,null,null,null,1] Output: 3 Explanation: Longest ZigZag path in blue nodes (right -> left -> right).

Example 2:

Input: root = [1,1,1,null,1,null,null,1,1,null,1] Output: 4 Explanation: Longest ZigZag path in blue nodes (left -> right -> left -> right).

Example 3: Input: root = [1] Output: 0

Constraints:

The number of nodes in the tree is in the range [1, 5 * 104]. 1 <= Node.val <= 100

==
Problem Number: 1120 URL: https://leetcode.com/problems/number-of-times-binary-string-is-prefix-aligned Title: 1375. Number of Times Binary String Is Prefix-Aligned Problem Description: You have a 1-indexed binary string of length n where all the bits are 0 initially. We will flip all the bits of this binary string (i.e., change them from 0 to 1) one by one. You are given a 1-indexed integer array flips where flips[i] indicates that the bit at index i will be flipped in the ith step. A binary string is prefix-aligned if, after the ith step, all the bits in the inclusive range [1, i] are ones and all the other bits are zeros. Return the number of times the binary string is prefix-aligned during the flipping process. Example 1: Input: flips = [3,2,4,1,5] Output: 2 Explanation: The binary string is initially "00000". After applying step 1: The string becomes "00100", which is not prefix-aligned. After applying step 2: The string becomes "01100", which is not prefix-aligned. After applying step 3: The string becomes "01110", which is not prefix-aligned. After applying step 4: The string becomes "11110", which is prefix-aligned. After applying step 5: The string becomes "11111", which is prefix-aligned. We can see that the string was prefix-aligned 2 times, so we return 2.

Example 2: Input: flips = [4,1,2,3] Output: 1 Explanation: The binary string is initially "0000". After applying step 1: The string becomes "0001", which is not prefix-aligned. After applying step 2: The string becomes "1001", which is not prefix-aligned. After applying step 3: The string becomes "1101", which is not prefix-aligned. After applying step 4: The string becomes "1111", which is prefix-aligned. We can see that the string was prefix-aligned 1 time, so we return 1.

Constraints:

n == flips.length 1 <= n <= 5 * 104 flips is a permutation of the integers in the range [1, n].

==
Problem Number: 1121 URL: https://leetcode.com/problems/time-needed-to-inform-all-employees Title: 1376. Time Needed to Inform All Employees Problem Description: A company has n employees with a unique ID for each employee from 0 to n - 1. The head of the company is the one with headID. Each employee has one direct manager given in the manager array where

manager[i] is the direct manager of the i-th employee, manager[headID] = -1. Also, it is guaranteed that the subordination relationships have a tree structure. The head of the company wants to inform all the company employees of an urgent piece of news. He will inform his direct subordinates, and they will inform their subordinates, and so on until all employees know about the urgent news. The i-th employee needs informTime[i] minutes to inform all of his direct subordinates (i.e., After informTime[i] minutes, all his direct subordinates can start spreading the news). Return the number of minutes needed to inform all the employees about the urgent news.   Example 1: Input: n = 1, headID = 0, manager = [-1], informTime = [0] Output: 0 Explanation: The head of the company is the only employee in the company.

Example 2:

Input: n = 6, headID = 2, manager = [2,2,-1,2,2,2], informTime = [0,0,1,0,0,0] Output: 1 Explanation: The head of the company with id = 2 is the direct manager of all the employees in the company and needs 1 minute to inform them all. The tree structure of the employees in the company is shown.

Constraints:

1 <= n <= 105 0 <= headID < n manager.length == n 0 <= manager[i] < n manager[headID] == -1 informTime.length == n 0 <= informTime[i] <= 1000 informTime[i] == 0 if employee i has no subordinates. It is guaranteed that all the employees can be informed.

==

Problem Number: 1122 URL: https://leetcode.com/problems/design-a-stack-with-increment-operation Title: 1381. Design a Stack With Increment Operation Problem Description: Design a stack that supports increment operations on its elements. Implement the CustomStack class:

CustomStack(int maxSize) Initializes the object with maxSize which is the maximum number of elements in the stack. void push(int x) Adds x to the top of the stack if the stack has not reached the maxSize. int pop() Pops and returns the top of the stack or -1 if the stack is empty. void inc(int k, int val) Increments the bottom k elements of the stack by val. If there are less than k elements in the stack, increment all the elements in the stack.

Example 1: Input ["CustomStack","push","push","pop","push","push","push","increment","increment","pop [[3],[1],[2],[],[2],[3],[4],[5,100],[2,100],[],[],[],[]] Output [null,null,null,2,null,null,null,null,null,103,202,201,-1] Explanation CustomStack stk = new CustomStack(3); // Stack is Empty [] stk.push(1); // stack becomes [1] stk.push(2); // stack becomes [1, 2] stk.pop(); // return 2 --> Return top of the stack 2, stack becomes [1] stk.push(2); // stack becomes [1, 2] stk.push(3); // stack becomes [1, 2, 3] stk.push(4); // stack still [1, 2, 3], Do not add another elements as size is 4 stk.increment(5, 100); // stack becomes [101, 102, 103] stk.increment(2, 100); // stack becomes [201, 202, 103] stk.pop(); // return 103 --> Return top of the stack 103, stack becomes [201, 202] stk.pop(); // return 202 --> Return top of the stack 202, stack becomes [201] stk.pop(); // return 201 --> Return top of the stack 201, stack becomes [] stk.pop(); // return -1 --> Stack is empty return -1.

Constraints:

1 <= maxSize, x, k <= 1000 0 <= val <= 100 At most 1000 calls will be made to each method of increment, push and pop each separately.

==
Problem Number: 1123 URL: https://leetcode.com/problems/balance-a-binary-search-tree Title: 1382. Balance a Binary Search Tree Problem Description: Given the root of a binary search tree, return a balanced binary search tree with the same node values. If there is more than one answer, return any of them. A binary search tree is balanced if the depth of the two subtrees of every node never differs by more than 1.   Example 1:

Input: root = [1,null,2,null,3,null,4,null,null] Output: [2,1,3,null,null,null,4] Explanation: This is not the only correct answer, [3,1,4,null,2] is also correct.

Example 2:

Input: root = [2,1,3] Output: [2,1,3]

Constraints:

The number of nodes in the tree is in the range [1, 104]. 1 <= Node.val <= 105

==
Problem Number: 1124 URL: https://leetcode.com/problems/cinema-seat-allocation Title: 1386. Cinema Seat Allocation Problem Description:

A cinema has n rows of seats, numbered from 1 to n and there are ten seats in each row, labelled from 1 to 10 as shown in the figure above. Given the array reservedSeats containing the numbers of seats already reserved, for example, reservedSeats[i] = [3,8] means the seat located in row 3 and labelled with 8 is already reserved. Return the maximum number of four-person groups you can assign on the cinema seats. A four-person group occupies four adjacent seats in one single row. Seats across an aisle (such as [3,3] and [3,4]) are not considered to be adjacent, but there is an exceptional case on which an aisle split a four-person group, in that case, the aisle split a four-person group in the middle, which means to have two people on each side.   Example 1:

Input: n = 3, reservedSeats = [[1,2],[1,3],[1,8],[2,6],[3,1],[3,10]] Output: 4 Explanation: The figure above shows the optimal allocation for four groups, where seats mark with blue are already reserved and contiguous seats mark with orange are for one group.

Example 2: Input: n = 2, reservedSeats = [[2,1],[1,8],[2,6]] Output: 2

Example 3: Input: n = 4, reservedSeats = [[4,3],[1,4],[4,6],[1,7]] Output: 4

Constraints:

1 <= n <= 10^9 1 <= reservedSeats.length <= min(10*n, 10^4) reservedSeats[i].length == 2 1 <= reservedSeats[i][0] <= n 1 <= reservedSeats[i][1] <= 10 All reservedSeats[i] are distinct.

==

Problem Number: 1125 URL: https://leetcode.com/problems/sort-integers-by-the-power-value Title: 1387. Sort Integers by The Power Value Problem Description: The power of an integer x is defined as the number of steps needed to transform x into 1 using the following steps:

if x is even then x = x / 2 if x is odd then x = 3 * x + 1

For example, the power of x = 3 is 7 because 3 needs 7 steps to become 1 (3 --> 10 --> 5 --> 16 --> 8 --> 4 --> 2 --> 1). Given three integers lo, hi and k. The task is to sort all integers in the interval [lo, hi] by the power value in ascending order, if two or more integers have the same power value sort them by ascending order. Return the kth integer in the range [lo, hi] sorted by the power value. Notice that for any integer x (lo <= x <= hi) it is guaranteed that x will transform into 1 using these steps and that the power of x is will fit in a 32-bit signed integer. Example 1: Input: lo = 12, hi = 15, k = 2 Output: 13 Explanation: The power of 12 is 9 (12 --> 6 --> 3 --> 10 --> 5 --> 16 --> 8 --> 4 --> 2 --> 1) The power of 13 is 9 The power of 14 is 17 The power of 15 is 17 The interval sorted by the power value [12,13,14,15]. For k = 2 answer is the second element which is 13. Notice that 12 and 13 have the same power value and we sorted them in ascending order. Same for 14 and 15.

Example 2: Input: lo = 7, hi = 11, k = 4 Output: 7 Explanation: The power array corresponding to the interval [7, 8, 9, 10, 11] is [16, 3, 19, 6, 14]. The interval sorted by power is [8, 10, 11, 7, 9]. The fourth number in the sorted array is 7.

Constraints:

1 <= lo <= hi <= 1000 1 <= k <= hi - lo + 1

==

Problem Number: 1126 URL: https://leetcode.com/problems/four-divisors Title: 1390. Four Divisors Problem Description: Given an integer array nums, return the sum of divisors of the integers in that array that have exactly four divisors. If there is no such integer in the array, return 0. Example 1: Input: nums = [21,4,7] Output: 32 Explanation: 21 has 4 divisors: 1, 3, 7, 21 4 has 3 divisors: 1, 2, 4 7 has 2 divisors: 1, 7 The answer is the sum of divisors of 21 only.

Example 2: Input: nums = [21,21] Output: 64

Example 3: Input: nums = [1,2,3,4,5] Output: 0

Constraints:

1 <= nums.length <= 104 1 <= nums[i] <= 105

==

Problem Number: 1127 URL: https://leetcode.com/problems/check-if-there-is-a-valid-path-in-a-grid Title: 1391. Check if There is a Valid Path in a Grid Problem Description: You are given an m x n grid. Each cell of grid represents a street. The street of grid[i][j] can be:

1 which means a street connecting the left cell and the right cell. 2 which means a street connecting the upper cell and the lower cell. 3 which means a street connecting the left cell and the lower cell. 4 which means a street connecting the right cell and the lower cell. 5 which means a street connecting the left cell and the upper cell. 6 which means a street connecting the right cell and the upper cell.

You will initially start at the street of the upper-left cell (0, 0). A valid path in the grid is a path that starts from the upper left cell (0, 0) and ends at the bottom-right cell (m - 1, n - 1). The path should only follow the streets. Notice that you are not allowed to change any street. Return true if there is a valid path in the grid or false otherwise.   Example 1:

Input: grid = [[2,4,3],[6,5,2]] Output: true Explanation: As shown you can start at cell (0, 0) and visit all the cells of the grid to reach (m - 1, n - 1).

Example 2:

Input: grid = [[1,2,1],[1,2,1]] Output: false Explanation: As shown you the street at cell (0, 0) is not connected with any street of any other cell and you will get stuck at cell (0, 0)

Example 3: Input: grid = [[1,1,2]] Output: false Explanation: You will get stuck at cell (0, 1) and you cannot reach cell (0, 2).

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 300 1 <= grid[i][j] <= 6

==

Problem Number: 1128 URL: https://leetcode.com/problems/count-number-of-teams Title: 1395. Count Number of Teams Problem Description: There are n soldiers standing in a line. Each soldier is assigned a unique rating value. You have to form a team of 3 soldiers amongst them under the following rules:

Choose 3 soldiers with index (i, j, k) with rating (rating[i], rating[j], rating[k]). A team is valid if: (rating[i] < rating[j] < rating[k]) or (rating[i] > rating[j] > rating[k]) where (0 <= i < j < k < n).

Return the number of teams you can form given the conditions. (soldiers can be part of multiple teams).   Example 1: Input: rating = [2,5,3,4,1] Output: 3 Explanation: We can form three teams given the conditions. (2,3,4), (5,4,1), (5,3,1).

Example 2: Input: rating = [2,1,3] Output: 0 Explanation: We can't form any team given the conditions.

Example 3: Input: rating = [1,2,3,4] Output: 4

Constraints:

n == rating.length 3 <= n <= 1000 1 <= rating[i] <= 105 All the integers in rating are unique.

===

Problem Number: 1129 URL: https://leetcode.com/problems/design-underground-system Title: 1396. Design Underground System Problem Description: An underground railway system is keeping track of customer travel times between different stations. They are using this data to calculate the average time it takes to travel from one station to another. Implement the UndergroundSystem class:

void checkIn(int id, string stationName, int t)

A customer with a card ID equal to id, checks in at the station stationName at time t. A customer can only be checked into one place at a time.

void checkOut(int id, string stationName, int t)

A customer with a card ID equal to id, checks out from the station stationName at time t.

double getAverageTime(string startStation, string endStation)

Returns the average time it takes to travel from startStation to endStation. The average time is computed from all the previous traveling times from startStation to endStation that happened directly, meaning a check in at startStation followed by a check out from endStation. The time it takes to travel from startStation to endStation may be different from the time it takes to travel from endStation to startStation. There will be at least one customer that has traveled from startStation to endStation before getAverageTime is called.

You may assume all calls to the checkIn and checkOut methods are consistent. If a customer checks in at time t1 then checks out at time t2, then t1 < t2. All events happen in chronological order. Example 1: Input ["UndergroundSystem","checkIn","checkIn","checkIn","checkOut","checkOut","checkOut","getAverageTime","getAverageTime" [[],[45,"Leyton",3],[32,"Paradise",8],[27,"Leyton",10],[45,"Waterloo",15],[27,"Waterloo",20],[32,"Cambridge",22

Output [null,null,null,null,null,null,null,14.00000,11.00000,null,11.00000,null,12.00000]

Explanation UndergroundSystem undergroundSystem = new UndergroundSystem(); undergroundSystem.checkIn(45, "Leyton", 3); undergroundSystem.checkIn(32, "Paradise", 8); undergroundSystem.checkIn(27, "Leyton", 10); undergroundSystem.checkOut(45, "Waterloo", 15); // Customer 45 "Leyton" -> "Waterloo" in 15-3 = 12 undergroundSystem.checkOut(27, "Waterloo", 20); // Customer 27 "Leyton" -> "Waterloo" in 20-10 = 10 undergroundSystem.checkOut(32, "Cambridge", 22); // Customer 32 "Paradise" -> "Cambridge" in 22-8 = 14 undergroundSystem.getAverageTime("Paradise", "Cambridge"); // return 14.00000. One trip "Paradise" -> "Cambridge",

(14) / 1 = 14 undergroundSystem.getAverageTime("Leyton", "Waterloo"); // return 11.00000. Two trips "Leyton" -> "Waterloo", (10 + 12) / 2 = 11 undergroundSystem.checkIn(10, "Leyton", 24); undergroundSystem.getAverageTime("Leyton", "Waterloo"); // return 11.00000 undergroundSystem.checkOut(10, "Waterloo", 38); // Customer 10 "Leyton" -> "Waterloo" in 38-24 = 14 undergroundSystem.getAverageTime("Leyton", "Waterloo"); // return 12.00000. Three trips "Leyton" -> "Waterloo", (10 + 12 + 14) / 3 = 12

Example 2: Input ["UndergroundSystem","checkIn","checkOut","getAverageTime","checkIn","checkOut","get
[[],[10,"Leyton",3],[10,"Paradise",8],["Leyton","Paradise"],[5,"Leyton",10],[5,"Paradise",16],["Leyton","Paradis

Output [null,null,null,5.00000,null,null,5.50000,null,null,6.66667]

Explanation UndergroundSystem undergroundSystem = new UndergroundSystem(); undergroundSystem.checkIn(10, "Leyton", 3); undergroundSystem.checkOut(10, "Paradise", 8); // Customer 10 "Leyton" -> "Paradise" in 8-3 = 5 undergroundSystem.getAverageTime("Leyton", "Paradise"); // return 5.00000, (5) / 1 = 5 undergroundSystem.checkIn(5, "Leyton", 10); undergroundSystem.checkOut(5, "Paradise", 16); // Customer 5 "Leyton" -> "Paradise" in 16-10 = 6 undergroundSystem.getAverageTime("Leyton", "Paradise"); // return 5.50000, (5 + 6) / 2 = 5.5 undergroundSystem.checkIn(2, "Leyton", 21); undergroundSystem.checkOut(2, "Paradise", 30); // Customer 2 "Leyton" -> "Paradise" in 30-21 = 9 undergroundSystem.getAverageTime("Leyton", "Paradise"); // return 6.66667, (5 + 6 + 9) / 3 = 6.66667

Constraints:

1 <= id, t <= 106 1 <= stationName.length, startStation.length, endStation.length <= 10 All strings consist of uppercase and lowercase English letters and digits. There will be at most 2 * 104 calls in total to checkIn, checkOut, and getAverageTime. Answers within 10-5 of the actual value will be accepted.

==
Problem Number: 1130 URL: https://leetcode.com/problems/construct-k-palindrome-strings Title: 1400. Construct K Palindrome Strings Problem Description: Given a string s and an integer k, return true if you can use all the characters in s to construct k palindrome strings or false otherwise. Example 1: Input: s = "annabelle", k = 2 Output: true Explanation: You can construct two palindromes using all characters in s. Some possible constructions "anna" + "elble", "anbna" + "elle", "anellena" + "b"

Example 2: Input: s = "leetcode", k = 3 Output: false Explanation: It is impossible to construct 3 palindromes using all the characters of s.

Example 3: Input: s = "true", k = 4 Output: true Explanation: The only possible solution is to put each character in a separate string.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters. 1 <= k <= 105

==
Problem Number: 1131 URL: https://leetcode.com/problems/circle-and-rectangle-overlapping Title: 1401. Circle and Rectangle Overlapping Problem Description: You are given a circle represented as (radius, xCenter, yCenter) and an axis-aligned rectangle represented as (x1, y1, x2, y2), where (x1, y1) are the coordinates of the bottom-left corner, and (x2, y2) are the coordinates of the top-right corner of the rectangle. Return true if the circle and rectangle are overlapped otherwise return false. In other words, check if there is any point (xi, yi) that belongs to the circle and the rectangle at the same time. Example 1:

Input: radius = 1, xCenter = 0, yCenter = 0, x1 = 1, y1 = -1, x2 = 3, y2 = 1 Output: true Explanation: Circle and rectangle share the point (1,0).

Example 2: Input: radius = 1, xCenter = 1, yCenter = 1, x1 = 1, y1 = -3, x2 = 2, y2 = -1 Output: false

Example 3:

Input: radius = 1, xCenter = 0, yCenter = 0, x1 = -1, y1 = 0, x2 = 0, y2 = 1 Output: true

Constraints:

1 <= radius <= 2000 -104 <= xCenter, yCenter <= 104 -104 <= x1 < x2 <= 104 -104 <= y1 < y2 <= 104

==
Problem Number: 1132 URL: https://leetcode.com/problems/number-of-steps-to-reduce-a-number-in-binary-representation-to-one Title: 1404. Number of Steps to Reduce a Number in Binary Representation to One Problem Description: Given the binary representation of an integer as a string s, return the number of steps to reduce it to 1 under the following rules:

If the current number is even, you have to divide it by 2.

If the current number is odd, you have to add 1 to it.

It is guaranteed that you can always reach one for all test cases. Example 1: Input: s = ”1101” Output: 6 Explanation: ”1101” corressponds to number 13 in their decimal representation. Step 1) 13 is odd, add 1 and obtain 14. Step 2) 14 is even, divide by 2 and obtain 7. Step 3) 7 is odd, add 1 and obtain 8. Step 4) 8 is even, divide by 2 and obtain 4. Step 5) 4 is even, divide by 2 and obtain 2. Step 6) 2 is even, divide by 2 and obtain 1.

Example 2: Input: s = ”10” Output: 1 Explanation: ”10” corressponds to number 2 in their decimal representation. Step 1) 2 is even, divide by 2 and obtain 1.

Example 3: Input: s = ”1” Output: 0

Constraints:

1 <= s.length <= 500 s consists of characters '0' or '1' s[0] == '1'

==
Problem Number: 1133 URL: https://leetcode.com/problems/longest-happy-string Title: 1405. Longest Happy String Problem Description: A string s is called happy if it satisfies the following conditions:

s only contains the letters 'a', 'b', and 'c'. s does not contain any of "aaa", "bbb", or "ccc" as a substring. s contains at most a occurrences of the letter 'a'. s contains at most b occurrences of the letter 'b'. s contains at most c occurrences of the letter 'c'.

Given three integers a, b, and c, return the longest possible happy string. If there are multiple longest happy strings, return any of them. If there is no such string, return the empty string "". A substring is a contiguous sequence of characters within a string. Example 1: Input: a = 1, b = 1, c = 7 Output: "ccaccbcc" Explanation: "ccbccacc" would also be a correct answer.

Example 2: Input: a = 7, b = 1, c = 0 Output: "aabaa" Explanation: It is the only correct answer in this case.

Constraints:

0 <= a, b, c <= 100 a + b + c > 0

==
Problem Number: 1134 URL: https://leetcode.com/problems/queries-on-a-permutation-with-key Title: 1409. Queries on a Permutation With Key Problem Description: Given the array queries of positive integers between 1 and m, you have to process all queries[i] (from i=0 to i=queries.length-1) according to the following rules:

In the beginning, you have the permutation P=[1,2,3,...,m]. For the current i, find the position of queries[i] in the permutation P (indexing from 0) and then move this at the beginning of the permutation P. Notice that the position of queries[i] in P is the result for queries[i].

Return an array containing the result for the given queries. Example 1: Input: queries = [3,1,2,1], m = 5 Output: [2,1,2,1] Explanation: The queries are processed as follow: For i=0: queries[i]=3, P=[1,2,3,4,5], position of 3 in P is 2, then we move 3 to the beginning of P resulting in P=[3,1,2,4,5]. For i=1: queries[i]=1, P=[3,1,2,4,5], position of 1 in P is 1, then we move 1 to the beginning of P resulting in P=[1,3,2,4,5]. For i=2: queries[i]=2, P=[1,3,2,4,5], position of 2 in P is 2, then we move 2 to the beginning of P resulting in P=[2,1,3,4,5]. For i=3: queries[i]=1, P=[2,1,3,4,5], position of 1 in P is 1, then we move 1 to the beginning of P resulting in P=[1,2,3,4,5]. Therefore, the array containing the result is [2,1,2,1].

Example 2: Input: queries = [4,1,2,2], m = 4 Output: [3,1,2,0]

Example 3: Input: queries = [7,5,5,8,3], m = 8 Output: [6,5,0,7,5]

Constraints:

1 <= m <= 10^3 1 <= queries.length <= m 1 <= queries[i] <= m

==
Problem Number: 1135 URL: https://leetcode.com/problems/html-entity-parser Title: 1410. HTML Entity Parser Problem Description: HTML entity parser is the parser that takes HTML code as input and replace all the entities of the special characters by the characters itself. The special characters and their entities for HTML are:

Quotation Mark: the entity is " and symbol character is ". Single Quote Mark: the entity is ' and symbol character is '. Ampersand: the entity is & and symbol character is &. Greater Than Sign: the entity is > and symbol character is >. Less Than Sign: the entity is < and symbol character is <. Slash: the entity is ⁄ and symbol character is /.

Given the input text string to the HTML parser, you have to implement the entity parser. Return the text after replacing the entities by the special characters. Example 1: Input: text = "& is an HTML entity but &ambassador; is not." Output: "& is an HTML entity but &ambassador; is not." Explanation: The parser will replace the & entity by &

Example 2: Input: text = "and I quote: "..."" Output: "and I quote: \"...\""

Constraints:

1 <= text.length <= 105 The string may contain any possible characters out of all the 256 ASCII characters.

==
Problem Number: 1136 URL: https://leetcode.com/problems/find-the-minimum-number-of-fibonacci-numbers-whose-sum-is-k Title: 1414. Find the Minimum Number of Fibonacci Numbers Whose Sum Is K Problem Description: Given an integer k, return the minimum number of Fibonacci numbers whose sum is equal to k. The same Fibonacci number can be used multiple times. The Fibonacci numbers are defined as:

F1 = 1 F2 = 1 Fn = Fn-1 + Fn-2 for n > 2.

It is guaranteed that for the given constraints we can always find such Fibonacci numbers that sum up to k. Example 1: Input: k = 7 Output: 2 Explanation: The Fibonacci numbers are: 1, 1, 2, 3, 5, 8, 13, ... For k = 7 we can use 2 + 5 = 7. Example 2: Input: k = 10 Output: 2 Explanation: For k = 10 we can use 2 + 8 = 10.

Example 3: Input: k = 19 Output: 3 Explanation: For k = 19 we can use 1 + 5 + 13 = 19.

Constraints:

1 <= k <= 109

==

Problem Number: 1137 URL: https://leetcode.com/problems/the-k-th-lexicographical-string-of-all-happy-strings-of-length-n Title: 1415. The k-th Lexicographical String of All Happy Strings of Length n Problem Description: A happy string is a string that:

consists only of letters of the set ['a', 'b', 'c']. s[i] != s[i + 1] for all values of i from 1 to s.length - 1 (string is 1-indexed).

For example, strings "abc", "ac", "b" and "abcbabcbcb" are all happy strings and strings "aa", "baa" and "ababbc" are not happy strings. Given two integers n and k, consider a list of all happy strings of length n sorted in lexicographical order. Return the kth string of this list or return an empty string if there are less than k happy strings of length n. Example 1: Input: n = 1, k = 3 Output: "c" Explanation: The list ["a", "b", "c"] contains all happy strings of length 1. The third string is "c".

Example 2: Input: n = 1, k = 4 Output: "" Explanation: There are only 3 happy strings of length 1.

Example 3: Input: n = 3, k = 9 Output: "cab" Explanation: There are 12 different happy string of length 3 ["aba", "abc", "aca", "acb", "bab", "bac", "bca", "bcb", "cab", "cac", "cba", "cbc"]. You will find the 9th string = "cab"

Constraints:

1 <= n <= 10 1 <= k <= 100

==

Problem Number: 1138 URL: https://leetcode.com/problems/display-table-of-food-orders-in-a-restaurant Title: 1418. Display Table of Food Orders in a Restaurant Problem Description: Given the array orders, which represents the orders that customers have done in a restaurant. More specifically orders[i]=[customerNamei,tableNumberi,foodItemi] where customerNamei is the name of the customer, tableNumberi is the table customer sit at, and foodItemi is the item customer orders. Return the restaurant's "display table". The "display table" is a table whose row entries denote how many of each food item each table ordered. The first column is the table number and the remaining columns correspond to each food item in alphabetical order. The first row should be a header whose first column is "Table", followed by the names of the food items. Note that the customer names are not part of the table. Additionally, the rows should be sorted in numerically increasing order. Example 1: Input: orders = [["David","3","Ceviche"],["Corina","10","Beef Burrito"],["David","3","Fried Chicken"],["Carla","5","Water"],["Carla","5","Ceviche"],["Rous","3","Ceviche"]] Output: [["Table","Beef Burrito","Ceviche","Fried Chicken","Water"],["3","0","2","1","0"],["5","0","1","0","1 Explanation: The displaying table looks like: Table,Beef Burrito,Ceviche,Fried Chicken,Water 3 ,0 ,2 ,1 ,0 5 ,0 ,1 ,0 ,1 10 ,1 ,0 ,0 ,0 For the table 3: David orders ”Ceviche” and ”Fried Chicken”, and Rous orders ”Ceviche”. For the table 5: Carla orders ”Water” and ”Ceviche”. For the table 10: Corina orders ”Beef Burrito”.

Example 2: Input: orders = [[”James”,”12”,”Fried Chicken”],[”Ratesh”,”12”,”Fried Chicken”],[”Amadeus”,”12”,”Fried Chicken”],[”Adam”,”1”,”Canadian Waffles”],[”Brianna”,”1”,”Canadian Waffles”]] Output: [[”Table”,”Canadian Waffles”,”Fried Chicken”],[”1”,”2”,”0”],[”12”,”0”,”3”]] Explanation: For the table 1: Adam and Brianna order ”Canadian Waffles”. For the table 12: James, Ratesh and Amadeus order ”Fried Chicken”.

Example 3: Input: orders = [[”Laura”,”2”,”Bean Burrito”],[”Jhon”,”2”,”Beef Burrito”],[”Melissa”,”2”,”Soda”]] Output: [[”Table”,”Bean Burrito”,”Beef Burrito”,”Soda”],[”2”,”1”,”1”,”1”]]

Constraints:

1 <= orders.length <= 5 * 10^4 orders[i].length == 3 1 <= customerNamei.length, foodItemi.length <= 20 customerNamei and foodItemi consist of lowercase and uppercase English letters and the space character. tableNumberi is a valid integer between 1 and 500.

==
Problem Number: 1139 URL: https://leetcode.com/problems/minimum-number-of-frogs-croaking Title: 1419. Minimum Number of Frogs Croaking Problem Description: You are given the string croakOfFrogs, which represents a combination of the string ”croak” from different frogs, that is, multiple frogs can croak at the same time, so multiple ”croak” are mixed. Return the minimum number of different frogs to finish all the croaks in the given string. A valid ”croak” means a frog is printing five letters ’c’, ’r’, ’o’, ’a’, and ’k’ sequentially. The frogs have to print all five letters to finish a croak. If the given string is not a combination of a valid ”croak” return -1. Example 1: Input: croakOfFrogs = ”croakcroak” Output: 1 Explanation: One frog yelling ”croak” twice.

Example 2: Input: croakOfFrogs = ”crcoakroak” Output: 2 Explanation: The minimum number of frogs is two. The first frog could yell ”crcoakroak”. The second frog could yell later ”crcoakroak”.

Example 3: Input: croakOfFrogs = ”croakcrook” Output: -1 Explanation: The given string is an invalid combination of ”croak” from different frogs.

Constraints:

1 <= croakOfFrogs.length <= 105 croakOfFrogs is either ’c’, ’r’, ’o’, ’a’, or ’k’.

==
Problem Number: 1140 URL: https://leetcode.com/problems/maximum-points-you-can-obtain-from-cards Title: 1423. Maximum Points You Can Obtain from Cards Problem Description: There are several cards arranged in a row, and each card has an associated number of points. The points are

given in the integer array cardPoints. In one step, you can take one card from the beginning or from the end of the row. You have to take exactly k cards. Your score is the sum of the points of the cards you have taken. Given the integer array cardPoints and the integer k, return the maximum score you can obtain. Example 1: Input: cardPoints = [1,2,3,4,5,6,1], k = 3 Output: 12 Explanation: After the first step, your score will always be 1. However, choosing the rightmost card first will maximize your total score. The optimal strategy is to take the three cards on the right, giving a final score of 1 + 6 + 5 = 12.

Example 2: Input: cardPoints = [2,2,2], k = 2 Output: 4 Explanation: Regardless of which two cards you take, your score will always be 4.

Example 3: Input: cardPoints = [9,7,7,9,7,7,9], k = 7 Output: 55 Explanation: You have to take all the cards. Your score is the sum of points of all cards.

Constraints:

1 <= cardPoints.length <= 105 1 <= cardPoints[i] <= 104 1 <= k <= cardPoints.length

==
Problem Number: 1141 URL: https://leetcode.com/problems/diagonal-traverse-ii Title: 1424. Diagonal Traverse II Problem Description: Given a 2D integer array nums, return all elements of nums in diagonal order as shown in the below images. Example 1:

Input: nums = [[1,2,3],[4,5,6],[7,8,9]] Output: [1,4,2,7,5,3,8,6,9]

Example 2:

Input: nums = [[1,2,3,4,5],[6,7],[8],[9,10,11],[12,13,14,15,16]] Output: [1,6,2,8,7,3,9,4,12,10,5,13,11,14,15,16]

Constraints:

1 <= nums.length <= 105 1 <= nums[i].length <= 105 1 <= sum(nums[i].length) <= 105 1 <= nums[i][j] <= 105

==
Problem Number: 1142 URL: https://leetcode.com/problems/max-difference-you-can-get-from-changing-an-integer Title: 1432. Max Difference You Can Get From Changing an Integer Problem Description: You are given an integer num. You will apply the following steps exactly two times:

Pick a digit x (0 <= x <= 9). Pick another digit y (0 <= y <= 9). The digit y can be equal to x. Replace all the occurrences of x in the decimal representation of num by y. The new integer cannot have any leading zeros, also the new integer cannot be 0.

Let a and b be the results of applying the operations to num the first and second times, respectively. Return the max difference between a and b. Example 1: Input: num = 555 Output: 888 Explanation: The first time pick x = 5 and y = 9 and store the new integer in a. The second time pick x = 5 and y = 1 and store the new integer in b. We have now a = 999 and b = 111 and max difference = 888

Example 2: Input: num = 9 Output: 8 Explanation: The first time pick x = 9 and y = 9 and store the new integer in a. The second time pick x = 9 and y = 1 and store the new integer in b. We have now a = 9 and b = 1 and max difference = 8

Constraints:

1 <= num <= 108

==

Problem Number: 1143 URL: https://leetcode.com/problems/check-if-a-string-can-break-another-string Title: 1433. Check If a String Can Break Another String Problem Description: Given two strings: s1 and s2 with the same size, check if some permutation of string s1 can break some permutation of string s2 or vice-versa. In other words s2 can break s1 or vice-versa. A string x can break string y (both of size n) if x[i] >= y[i] (in alphabetical order) for all i between 0 and n-1. Example 1: Input: s1 = "abc", s2 = "xya" Output: true Explanation: "ayx" is a permutation of s2="xya" which can break to string "abc" which is a permutation of s1="abc".

Example 2: Input: s1 = "abe", s2 = "acd" Output: false Explanation: All permutations for s1="abe" are: "abe", "aeb", "bae", "bea", "eab" and "eba" and all permutation for s2="acd" are: "acd", "adc", "cad", "cda", "dac" and "dca". However, there is not any permutation from s1 which can break some permutation from s2 and vice-versa.

Example 3: Input: s1 = "leetcodee", s2 = "interview" Output: true

Constraints:

s1.length == n s2.length == n 1 <= n <= 10^5 All strings consist of lowercase English letters.

==

Problem Number: 1144 URL: https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit Title: 1438. Longest Continuous Subarray With Absolute Diff Less Than or Equal to Limit Problem Description: Given an array of integers nums and an integer limit, return the size of the longest non-empty subarray such that the absolute difference between any two elements of this subarray is less than or equal to limit. Example 1: Input: nums = [8,2,4,7], limit = 4 Output: 2 Explanation: All subarrays are: [8] with maximum absolute diff |8-8| = 0 <= 4. [8,2] with maximum absolute diff |8-2| = 6 > 4. [8,2,4] with maximum absolute diff |8-2| = 6 > 4. [8,2,4,7] with maximum absolute diff |8-2| = 6 > 4. [2] with maximum absolute diff |2-2| = 0 <= 4. [2,4] with maximum absolute diff |2-4| = 2 <= 4. [2,4,7] with maximum absolute diff |2-7| = 5 > 4. [4] with maximum absolute diff |4-4| = 0 <= 4. [4,7] with maximum absolute diff |4-7| = 3 <= 4. [7] with maximum absolute diff |7-7| = 0 <= 4. Therefore, the size of the longest subarray is 2.

Example 2: Input: nums = [10,1,2,4,7,2], limit = 5 Output: 4 Explanation: The subarray [2,4,7,2] is the longest since the maximum absolute diff is |2-7| = 5 <= 5.

Example 3: Input: nums = [4,2,2,2,4,4,2,2], limit = 0 Output: 3

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 0 <= limit <= 109

==
Problem Number: 1145 URL: https://leetcode.com/problems/build-an-array-with-stack-operations Title: 1441. Build an Array With Stack Operations Problem Description: You are given an integer array target and an integer n. You have an empty stack with the two following operations:

"Push": pushes an integer to the top of the stack. "Pop": removes the integer on the top of the stack.

You also have a stream of the integers in the range [1, n]. Use the two stack operations to make the numbers in the stack (from the bottom to the top) equal to target. You should follow the following rules:

If the stream of the integers is not empty, pick the next integer from the stream and push it to the top of the stack. If the stack is not empty, pop the integer at the top of the stack. If, at any moment, the elements in the stack (from the bottom to the top) are equal to target, do not read new integers from the stream and do not do more operations on the stack.

Return the stack operations needed to build target following the mentioned rules. If there are multiple valid answers, return any of them. Example 1: Input: target = [1,3], n = 3 Output: ["Push","Push","Pop","Push"] Explanation: Initially the stack s is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack. s = [1]. Read 2 from the stream and push it to the stack. s = [1,2]. Pop the integer on the top of the stack. s = [1]. Read 3 from the stream and push it to the stack. s = [1,3].

Example 2: Input: target = [1,2,3], n = 3 Output: ["Push","Push","Push"] Explanation: Initially the stack s is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack. s = [1]. Read 2 from the stream and push it to the stack. s = [1,2]. Read 3 from the stream and push it to the stack. s = [1,2,3].

Example 3: Input: target = [1,2], n = 4 Output: ["Push","Push"] Explanation: Initially the stack s is empty. The last element is the top of the stack. Read 1 from the stream and push it to the stack. s = [1]. Read 2 from the stream and push it to the stack. s = [1,2]. Since the stack (from the bottom to the top) is equal to target, we stop the stack operations. The answers that read integer 3 from the stream are not accepted.

Constraints:

1 <= target.length <= 100 1 <= n <= 100 1 <= target[i] <= n target is strictly increasing.

==

Problem Number: 1146 URL: https://leetcode.com/problems/count-triplets-that-can-form-two-arrays-of-equal-xor Title: 1442. Count Triplets That Can Form Two Arrays of Equal XOR Problem Description: Given an array of integers arr. We want to select three indices i, j and k where (0 <= i < j <= k < arr.length). Let's define a and b as follows:

a = arr[i] ^ arr[i + 1] ^ ... ^ arr[j - 1] b = arr[j] ^ arr[j + 1] ^ ... ^ arr[k]

Note that ^ denotes the bitwise-xor operation. Return the number of triplets (i, j and k) Where a == b. Example 1: Input: arr = [2,3,1,6,7] Output: 4 Explanation: The triplets are (0,1,2), (0,2,2), (2,3,4) and (2,4,4)

Example 2: Input: arr = [1,1,1,1,1] Output: 10

Constraints:

1 <= arr.length <= 300 1 <= arr[i] <= 108

==

Problem Number: 1147 URL: https://leetcode.com/problems/minimum-time-to-collect-all-apples-in-a-tree Title: 1443. Minimum Time to Collect All Apples in a Tree Problem Description: Given an undirected tree consisting of n vertices numbered from 0 to n-1, which has some apples in their vertices. You spend 1 second to walk over one edge of the tree. Return the minimum time in seconds you have to spend to collect all apples in the tree, starting at vertex 0 and coming back to this vertex. The edges of the undirected tree are given in the array edges, where edges[i] = [ai, bi] means that exists an edge connecting the vertices ai and bi. Additionally, there is a boolean array hasApple, where hasApple[i] = true means that vertex i has an apple; otherwise, it does not have any apple. Example 1:

Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,true,false,true,true,false] Output: 8 Explanation: The figure above represents the given tree where red vertices have an apple. One optimal path to collect all apples is shown by the green arrows.

Example 2:

Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,true,false,false,true,false] Output: 6 Explanation: The figure above represents the given tree where red vertices have an apple. One optimal path to collect all apples is shown by the green arrows.

Example 3: Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], hasApple = [false,false,false,false,false,false,false] Output: 0

Constraints:

1 <= n <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ai < bi <= n - 1 hasApple.length == n

==
Problem Number: 1148 URL: https://leetcode.com/problems/simplified-fractions Title: 1447. Simplified Fractions Problem Description: Given an integer n, return a list of all simplified fractions between 0 and 1 (exclusive) such that the denominator is less-than-or-equal-to n. You can return the answer in any order. Example 1: Input: n = 2 Output: ["1/2"] Explanation: "1/2" is the only unique fraction with a denominator less-than-or-equal-to 2.

Example 2: Input: n = 3 Output: ["1/2","1/3","2/3"]

Example 3: Input: n = 4 Output: ["1/2","1/3","1/4","2/3","3/4"] Explanation: "2/4" is not a simplified fraction because it can be simplified to "1/2".

Constraints:

1 <= n <= 100

==
Problem Number: 1149 URL: https://leetcode.com/problems/count-good-nodes-in-binary-tree Title: 1448. Count Good Nodes in Binary Tree Problem Description: Given a binary tree root, a node X in the tree is named good if in the path from root to X there are no nodes with a value greater than X. Return the number of good nodes in the binary tree. Example 1:

Input: root = [3,1,4,3,null,1,5] Output: 4 Explanation: Nodes in blue are good. Root Node (3) is always a good node. Node 4 -> (3,4) is the maximum value in the path starting from the root. Node 5 -> (3,4,5) is the maximum value in the path Node 3 -> (3,1,3) is the maximum value in the path. Example 2:

Input: root = [3,3,null,4,2] Output: 3 Explanation: Node 2 -> (3, 3, 2) is not good, because "3" is higher than it. Example 3: Input: root = [1] Output: 1 Explanation: Root is considered as good. Constraints:

The number of nodes in the binary tree is in the range [1, 10^5]. Each node's value is between [-10^4, 10^4].

==
Problem Number: 1150 URL: https://leetcode.com/problems/rearrange-words-in-a-sentence Title: 1451. Rearrange Words in a Sentence Problem Description: Given a sentence text (A sentence is a string of space-separated words) in the following format:

First letter is in upper case. Each word in text are separated by a single space.

Your task is to rearrange the words in text such that all words are rearranged in an increasing order of their lengths. If two words have the same length, arrange them in their original order. Return the new text following the format shown above. Example 1: Input: text = ”Leetcode is cool” Output: ”Is cool leetcode” Explanation: There are 3 words, ”Leetcode” of length 8, ”is” of length 2 and ”cool” of length 4. Output is ordered by length and the new first word starts with capital letter.

Example 2: Input: text = ”Keep calm and code on” Output: ”On and keep calm code” Explanation: Output is ordered as follows: ”On” 2 letters. ”and” 3 letters. ”keep” 4 letters in case of tie order by position in original text. ”calm” 4 letters. ”code” 4 letters.

Example 3: Input: text = ”To be or not to be” Output: ”To be or to be not”

Constraints:

text begins with a capital letter and then contains lowercase letters and single space between words. 1 <= text.length <= 10^5

==
Problem Number: 1151 URL: https://leetcode.com/problems/people-whose-list-of-favorite-companies-is-not-a-subset-of-another-list Title: 1452. People Whose List of Favorite Companies Is Not a Subset of Another List Problem Description: Given the array favoriteCompanies where favoriteCompanies[i] is the list of favorites companies for the ith person (indexed from 0). Return the indices of people whose list of favorite companies is not a subset of any other list of favorites companies. You must return the indices in increasing order. Example 1: Input: favoriteCompanies = [[”leetcode”,”google”,”facebook”],[”google”,”microsoft”],[”google”,”facebook”],[”google”],[”amazon”]] Output: [0,1,4] Explanation: Person with index=2 has favoriteCompanies[2]=[”google”,”facebook”] which is a subset of favoriteCompanies[0]=[”leetcode”,”google”,”facebook”] corresponding to the person with index 0. Person with index=3 has favoriteCompanies[3]=[”google”] which is a subset of favoriteCompanies[0]=[”leetcode”,”google”,”facebook”] and favoriteCompanies[1]=[”google”,”microsoft”]. Other lists of favorite companies are not a subset of another list, therefore, the answer is [0,1,4].

Example 2: Input: favoriteCompanies = [[”leetcode”,”google”,”facebook”],[”leetcode”,”amazon”],[”facebook”,”
Output: [0,1] Explanation: In this case favoriteCompanies[2]=[”facebook”,”google”] is a subset of favoriteCompanies[0]=[”leetcode”,”google”,”facebook”], therefore, the answer is [0,1].

Example 3: Input: favoriteCompanies = [[”leetcode”],[”google”],[”facebook”],[”amazon”]] Output: [0,1,2,3]

Constraints:

1 <= favoriteCompanies.length <= 100 1 <= favoriteCompanies[i].length <= 500 1 <= favoriteCompanies[i][j].length <= 20 All strings in favoriteCompa-

nies[i] are distinct. All lists of favorite companies are distinct, that is, If we sort alphabetically each list then favoriteCompanies[i] != favoriteCompanies[j]. All strings consist of lowercase English letters only.

==
Problem Number: 1152 URL: https://leetcode.com/problems/maximum-number-of-vowels-in-a-substring-of-given-length Title: 1456. Maximum Number of Vowels in a Substring of Given Length Problem Description: Given a string s and an integer k, return the maximum number of vowel letters in any substring of s with length k. Vowel letters in English are 'a', 'e', 'i', 'o', and 'u'. Example 1: Input: s = "abciiidef", k = 3 Output: 3 Explanation: The substring "iii" contains 3 vowel letters.

Example 2: Input: s = "aeiou", k = 2 Output: 2 Explanation: Any substring of length 2 contains 2 vowels.

Example 3: Input: s = "leetcode", k = 3 Output: 2 Explanation: "lee", "eet" and "ode" contain 2 vowels.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters. 1 <= k <= s.length

==
Problem Number: 1153 URL: https://leetcode.com/problems/pseudo-palindromic-paths-in-a-binary-tree Title: 1457. Pseudo-Palindromic Paths in a Binary Tree Problem Description: Given a binary tree where node values are digits from 1 to 9. A path in the binary tree is said to be pseudo-palindromic if at least one permutation of the node values in the path is a palindrome. Return the number of pseudo-palindromic paths going from the root node to leaf nodes. Example 1:

Input: root = [2,3,1,3,1,null,1] Output: 2 Explanation: The figure above represents the given binary tree. There are three paths going from the root node to leaf nodes: the red path [2,3,3], the green path [2,1,1], and the path [2,3,1]. Among these paths only red path and green path are pseudo-palindromic paths since the red path [2,3,3] can be rearranged in [3,2,3] (palindrome) and the green path [2,1,1] can be rearranged in [1,2,1] (palindrome).

Example 2:

Input: root = [2,1,1,1,3,null,null,null,null,null,1] Output: 1 Explanation: The figure above represents the given binary tree. There are three paths going from the root node to leaf nodes: the green path [2,1,1], the path [2,1,3,1], and the path [2,1]. Among these paths only the green path is pseudo-palindromic since [2,1,1] can be rearranged in [1,2,1] (palindrome).

Example 3: Input: root = [9] Output: 1

Constraints:

The number of nodes in the tree is in the range [1, 105]. 1 <= Node.val <= 9

==

Problem Number: 1154 URL: https://leetcode.com/problems/check-if-a-string-contains-all-binary-codes-of-size-k Title: 1461. Check If a String Contains All Binary Codes of Size K Problem Description: Given a binary string s and an integer k, return true if every binary code of length k is a substring of s. Otherwise, return false.   Example 1: Input: s = "00110110", k = 2 Output: true Explanation: The binary codes of length 2 are "00", "01", "10" and "11". They can be all found as substrings at indices 0, 1, 3 and 2 respectively.

Example 2: Input: s = "0110", k = 1 Output: true Explanation: The binary codes of length 1 are "0" and "1", it is clear that both exist as a substring.

Example 3: Input: s = "0110", k = 2 Output: false Explanation: The binary code "00" is of length 2 and does not exist in the array.

Constraints:

1 <= s.length <= 5 * 105 s[i] is either '0' or '1'. 1 <= k <= 20

==

Problem Number: 1155 URL: https://leetcode.com/problems/course-schedule-iv Title: 1462. Course Schedule IV Problem Description: There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [ai, bi] indicates that you must take course ai first if you want to take course bi.

For example, the pair [0, 1] indicates that you have to take course 0 before you can take course 1.

Prerequisites can also be indirect. If course a is a prerequisite of course b, and course b is a prerequisite of course c, then course a is a prerequisite of course c. You are also given an array queries where queries[j] = [uj, vj]. For the jth query, you should answer whether course uj is a prerequisite of course vj or not. Return a boolean array answer, where answer[j] is the answer to the jth query. Example 1:

Input: numCourses = 2, prerequisites = [[1,0]], queries = [[0,1],[1,0]] Output: [false,true] Explanation: The pair [1, 0] indicates that you have to take course 1 before you can take course 0. Course 0 is not a prerequisite of course 1, but the opposite is true.

Example 2: Input: numCourses = 2, prerequisites = [], queries = [[1,0],[0,1]] Output: [false,false] Explanation: There are no prerequisites, and each course is independent.

Example 3:

Input: numCourses = 3, prerequisites = [[1,2],[1,0],[2,0]], queries = [[1,0],[1,2]] Output: [true,true]

Constraints:

2 <= numCourses <= 100 0 <= prerequisites.length <= (numCourses * (numCourses - 1) / 2) prerequisites[i].length == 2 0 <= ai, bi <= n - 1 ai != bi All the pairs [ai, bi] are unique. The prerequisites graph has no cycles. 1 <= queries.length <= 104 0 <= ui, vi <= n - 1 ui != vi

==
Problem Number: 1156 URL: https://leetcode.com/problems/maximum-area-of-a-piece-of-cake-after-horizontal-and-vertical-cuts Title: 1465. Maximum Area of a Piece of Cake After Horizontal and Vertical Cuts Problem Description: You are given a rectangular cake of size h x w and two arrays of integers horizontalCuts and verticalCuts where:

horizontalCuts[i] is the distance from the top of the rectangular cake to the ith horizontal cut and similarly, and verticalCuts[j] is the distance from the left of the rectangular cake to the jth vertical cut.

Return the maximum area of a piece of cake after you cut at each horizontal and vertical position provided in the arrays horizontalCuts and verticalCuts. Since the answer can be a large number, return this modulo 109 + 7. Example 1:

Input: h = 5, w = 4, horizontalCuts = [1,2,4], verticalCuts = [1,3] Output: 4 Explanation: The figure above represents the given rectangular cake. Red lines are the horizontal and vertical cuts. After you cut the cake, the green piece of cake has the maximum area.

Example 2:

Input: h = 5, w = 4, horizontalCuts = [3,1], verticalCuts = [1] Output: 6 Explanation: The figure above represents the given rectangular cake. Red lines are the horizontal and vertical cuts. After you cut the cake, the green and yellow pieces of cake have the maximum area.

Example 3: Input: h = 5, w = 4, horizontalCuts = [3], verticalCuts = [3] Output: 9

Constraints:

2 <= h, w <= 109 1 <= horizontalCuts.length <= min(h - 1, 105) 1 <= verticalCuts.length <= min(w - 1, 105) 1 <= horizontalCuts[i] < h 1 <= verticalCuts[i] < w All the elements in horizontalCuts are distinct. All the elements in verticalCuts are distinct.

==
Problem Number: 1157 URL: https://leetcode.com/problems/reorder-routes-to-make-all-paths-lead-to-the-city-zero Title: 1466. Reorder Routes to Make All Paths Lead to the City Zero Problem Description: There are n cities numbered from 0 to n - 1 and n - 1 roads such that there is only one way to travel between two different cities (this network form a tree). Last year, The ministry of transport decided to orient the roads in one direction because they

are too narrow. Roads are represented by connections where connections[i] = [ai, bi] represents a road from city ai to city bi. This year, there will be a big event in the capital (city 0), and many people want to travel to this city. Your task consists of reorienting some roads such that each city can visit the city 0. Return the minimum number of edges changed. It's guaranteed that each city can reach city 0 after reorder.   Example 1:

Input: n = 6, connections = [[0,1],[1,3],[2,3],[4,0],[4,5]] Output: 3 Explanation: Change the direction of edges show in red such that each node can reach the node 0 (capital).

Example 2:

Input: n = 5, connections = [[1,0],[1,2],[3,2],[3,4]] Output: 2 Explanation: Change the direction of edges show in red such that each node can reach the node 0 (capital).

Example 3: Input: n = 3, connections = [[1,0],[2,0]] Output: 0

Constraints:

2 <= n <= 5 * 104 connections.length == n - 1 connections[i].length == 2 0 <= ai, bi <= n - 1 ai != bi

==
Problem Number: 1158 URL: https://leetcode.com/problems/the-k-strongest-values-in-an-array Title: 1471. The k Strongest Values in an Array Problem Description: Given an array of integers arr and an integer k. A value arr[i] is said to be stronger than a value arr[j] if |arr[i] - m| > |arr[j] - m| where m is the median of the array. If |arr[i] - m| == |arr[j] - m|, then arr[i] is said to be stronger than arr[j] if arr[i] > arr[j]. Return a list of the strongest k values in the array. return the answer in any arbitrary order. Median is the middle value in an ordered integer list. More formally, if the length of the list is n, the median is the element in position ((n - 1) / 2) in the sorted list (0-indexed).

For arr = [6, -3, 7, 2, 11], n = 5 and the median is obtained by sorting the array arr = [-3, 2, 6, 7, 11] and the median is arr[m] where m = ((5 - 1) / 2) = 2. The median is 6. For arr = [-7, 22, 17, 3], n = 4 and the median is obtained by sorting the array arr = [-7, 3, 17, 22] and the median is arr[m] where m = ((4 - 1) / 2) = 1. The median is 3.

Example 1: Input: arr = [1,2,3,4,5], k = 2 Output: [5,1] Explanation: Median is 3, the elements of the array sorted by the strongest are [5,1,4,2,3]. The strongest 2 elements are [5, 1]. [1, 5] is also accepted answer. Please note that although |5 - 3| == |1 - 3| but 5 is stronger than 1 because 5 > 1.

Example 2: Input: arr = [1,1,3,5,5], k = 2 Output: [5,5] Explanation: Median is 3, the elements of the array sorted by the strongest are [5,5,1,1,3]. The strongest 2 elements are [5, 5].

Example 3: Input: arr = [6,7,11,7,6,8], k = 5 Output: [11,8,6,6,7] Explanation:

Median is 7, the elements of the array sorted by the strongest are [11,8,6,6,7,7]. Any permutation of [11,8,6,6,7] is accepted.

Constraints:

1 <= arr.length <= 105 -105 <= arr[i] <= 105 1 <= k <= arr.length

==

Problem Number: 1159 URL: https://leetcode.com/problems/design-browser-history Title: 1472. Design Browser History Problem Description: You have a browser of one tab where you start on the homepage and you can visit another url, get back in the history number of steps or move forward in the history number of steps. Implement the BrowserHistory class:

BrowserHistory(string homepage) Initializes the object with the homepage of the browser. void visit(string url) Visits url from the current page. It clears up all the forward history. string back(int steps) Move steps back in history. If you can only return x steps in the history and steps > x, you will return only x steps. Return the current url after moving back in history at most steps. string forward(int steps) Move steps forward in history. If you can only forward x steps in the history and steps > x, you will forward only x steps. Return the current url after forwarding in history at most steps.

Example: Input: ["BrowserHistory","visit","visit","visit","back","back","forward","visit","forward","back","
[["leetcode.com"],["google.com"],["facebook.com"],["youtube.com"],[1],[1],[1],["linkedin.com"],[2],[2],[7]]
Output: [null,null,null,null,"facebook.com","google.com","facebook.com",null,"linkedin.com","google.com","le

Explanation: BrowserHistory browserHistory = new BrowserHistory("leetcode.com"); browserHistory.visit("google.com"); // You are in "leetcode.com". Visit "google.com" browserHistory.visit("facebook.com"); // You are in "google.com". Visit "facebook.com" browserHistory.visit("youtube.com"); // You are in "facebook.com". Visit "youtube.com" browserHistory.back(1); // You are in "youtube.com", move back to "facebook.com" return "facebook.com" browserHistory.back(1); // You are in "facebook.com", move back to "google.com" return "google.com" browserHistory.forward(1); // You are in "google.com", move forward to "facebook.com" return "facebook.com" browserHistory.visit("linkedin.com"); // You are in "facebook.com". Visit "linkedin.com" browserHistory.forward(2); // You are in "linkedin.com", you cannot move forward any steps. browserHistory.back(2); // You are in "linkedin.com", move back two steps to "facebook.com" then to "google.com". return "google.com" browserHistory.back(7); // You are in "google.com", you can move back only one step to "leetcode.com". return "leetcode.com"

Constraints:

1 <= homepage.length <= 20 1 <= url.length <= 20 1 <= steps <= 100 homepage and url consist of '.' or lower case English letters. At most 5000 calls will be made to visit, back, and forward.

==

Problem Number: 1160 URL: https://leetcode.com/problems/subrectangle-queries Title: 1476. Subrectangle Queries Problem Description: Implement the class SubrectangleQueries which receives a rows x cols rectangle as a matrix of integers in the constructor and supports two methods: 1. updateSubrectangle(int row1, int col1, int row2, int col2, int newValue)

Updates all values with newValue in the subrectangle whose upper left coordinate is (row1,col1) and bottom right coordinate is (row2,col2).

2. getValue(int row, int col)

Returns the current value of the coordinate (row,col) from the rectangle.

Example 1: Input ["SubrectangleQueries","getValue","updateSubrectangle","getValue","getValue","updateSu
[[[[1,2,1],[4,3,4],[3,2,1],[1,1,1]]],[0,2],[0,0,3,2,5],[0,2],[3,1],[3,0,3,2,10],[3,1],[0,2]]
Output [null,1,null,5,5,null,10,5] Explanation SubrectangleQueries subrectangleQueries = new SubrectangleQueries([[1,2,1],[4,3,4],[3,2,1],[1,1,1]]); // The initial rectangle (4x3) looks like: // 1 2 1 // 4 3 4 // 3 2 1 // 1 1 1 subrectangleQueries.getValue(0, 2); // return 1 subrectangleQueries.updateSubrectangle(0, 0, 3, 2, 5); // After this update the rectangle looks like: // 5 5 5 // 5 5 5 // 5 5 5 // 5 5 5 subrectangleQueries.getValue(0, 2); // return 5 subrectangleQueries.getValue(3, 1); // return 5 subrectangleQueries.updateSubrectangle(3, 0, 3, 2, 10); // After this update the rectangle looks like: // 5 5 5 // 5 5 5 // 5 5 5 // 10 10 10 subrectangleQueries.getValue(3, 1); // return 10 subrectangleQueries.getValue(0, 2); // return 5

Example 2: Input ["SubrectangleQueries","getValue","updateSubrectangle","getValue","getValue","updateSub
[[[[1,1,1],[2,2,2],[3,3,3]]],[0,0],[0,0,2,2,100],[0,0],[2,2],[1,1,2,2,20],[2,2]] Output [null,1,null,100,100,null,20] Explanation SubrectangleQueries subrectangleQueries = new SubrectangleQueries([[1,1,1],[2,2,2],[3,3,3]]); subrectangleQueries.getValue(0, 0); // return 1 subrectangleQueries.updateSubrectangle(0, 0, 2, 2, 100); subrectangleQueries.getValue(0, 0); // return 100 subrectangleQueries.getValue(2, 2); // return 100 subrectangleQueries.updateSubrectangle(1, 1, 2, 2, 20); subrectangleQueries.getValue(2, 2); // return 20

Constraints:

There will be at most 500 operations considering both methods: updateSubrectangle and getValue. 1 <= rows, cols <= 100 rows == rectangle.length cols == rectangle[i].length 0 <= row1 <= row2 < rows 0 <= col1 <= col2 < cols 1 <= newValue, rectangle[i][j] <= 10^9 0 <= row < rows 0 <= col < cols

==
Problem Number: 1161 URL: https://leetcode.com/problems/find-two-non-overlapping-sub-arrays-each-with-target-sum Title: 1477. Find Two Non-overlapping Sub-arrays Each With Target Sum Problem Description: You are given an array of integers arr and an integer target. You have to find two non-overlapping sub-arrays of arr each with a sum equal target. There can be multiple answers so you have to find an answer where the sum of the lengths of the two sub-arrays is minimum. Return the minimum sum of the lengths of

the two required sub-arrays, or return -1 if you cannot find such two sub-arrays. Example 1: Input: arr = [3,2,2,4,3], target = 3 Output: 2 Explanation: Only two sub-arrays have sum = 3 ([3] and [3]). The sum of their lengths is 2.

Example 2: Input: arr = [7,3,4,7], target = 7 Output: 2 Explanation: Although we have three non-overlapping sub-arrays of sum = 7 ([7], [3,4] and [7]), but we will choose the first and third sub-arrays as the sum of their lengths is 2.

Example 3: Input: arr = [4,3,2,6,2,3,4], target = 6 Output: -1 Explanation: We have only one sub-array of sum = 6.

Constraints:

1 <= arr.length <= 105 1 <= arr[i] <= 1000 1 <= target <= 108

== Problem Number: 1162 URL: https://leetcode.com/problems/least-number-of-unique-integers-after-k-removals Title: 1481. Least Number of Unique Integers after K Removals Problem Description: Given an array of integers arr and an integer k. Find the least number of unique integers after removing exactly k elements.

Example 1: Input: arr = [5,5,4], k = 1 Output: 1 Explanation: Remove the single 4, only 5 is left.

Example 2: Input: arr = [4,3,1,1,3,3,2], k = 3 Output: 2 Explanation: Remove 4, 2 and either one of the two 1s or three 3s. 1 and 3 will be left. Constraints:

1 <= arr.length <= 10^5 1 <= arr[i] <= 10^9 0 <= k <= arr.length

== Problem Number: 1163 URL: https://leetcode.com/problems/minimum-number-of-days-to-make-m-bouquets Title: 1482. Minimum Number of Days to Make m Bouquets Problem Description: You are given an integer array bloomDay, an integer m and an integer k. You want to make m bouquets. To make a bouquet, you need to use k adjacent flowers from the garden. The garden consists of n flowers, the ith flower will bloom in the bloomDay[i] and then can be used in exactly one bouquet. Return the minimum number of days you need to wait to be able to make m bouquets from the garden. If it is impossible to make m bouquets return -1. Example 1: Input: bloomDay = [1,10,3,10,2], m = 3, k = 1 Output: 3 Explanation: Let us see what happened in the first three days. x means flower bloomed and _ means flower did not bloom in the garden. We need 3 bouquets each should contain 1 flower. After day 1: [x, _, _, _, _] // we can only make one bouquet. After day 2: [x, _, _, _, x] // we can only make two bouquets. After day 3: [x, _, x, _, x] // we can make 3 bouquets. The answer is 3.

Example 2: Input: bloomDay = [1,10,3,10,2], m = 3, k = 2 Output: -1 Explanation: We need 3 bouquets each has 2 flowers, that means we need 6 flowers. We only have 5 flowers so it is impossible to get the needed bouquets and we return -1.

Example 3: Input: bloomDay = [7,7,7,7,12,7,7], m = 2, k = 3 Output: 12 Explanation: We need 2 bouquets each should have 3 flowers. Here is the garden after the 7 and 12 days: After day 7: [x, x, x, x, _, x, x] We can make one bouquet of the first three flowers that bloomed. We cannot make another bouquet from the last three flowers that bloomed because they are not adjacent. After day 12: [x, x, x, x, x, x, x] It is obvious that we can make two bouquets in different ways.

Constraints:

bloomDay.length == n 1 <= n <= 105 1 <= bloomDay[i] <= 109 1 <= m <= 106 1 <= k <= n

==
Problem Number: 1164 URL: https://leetcode.com/problems/making-file-names-unique Title: 1487. Making File Names Unique Problem Description: Given an array of strings names of size n. You will create n folders in your file system such that, at the ith minute, you will create a folder with the name names[i]. Since two files cannot have the same name, if you enter a folder name that was previously used, the system will have a suffix addition to its name in the form of (k), where, k is the smallest positive integer such that the obtained name remains unique. Return an array of strings of length n where ans[i] is the actual name the system will assign to the ith folder when you create it. Example 1: Input: names = ["pes","fifa","gta","pes(2019)"] Output: ["pes","fifa","gta","pes(2019)"] Explanation: Let's see how the file system creates folder names: "pes" --> not assigned before, remains "pes" "fifa" --> not assigned before, remains "fifa" "gta" --> not assigned before, remains "gta" "pes(2019)" --> not assigned before, remains "pes(2019)"

Example 2: Input: names = ["gta","gta(1)","gta","avalon"] Output: ["gta","gta(1)","gta(2)","avalon"] Explanation: Let's see how the file system creates folder names: "gta" --> not assigned before, remains "gta" "gta(1)" --> not assigned before, remains "gta(1)" "gta" --> the name is reserved, system adds (k), since "gta(1)" is also reserved, systems put k = 2. it becomes "gta(2)" "avalon" --> not assigned before, remains "avalon"

Example 3: Input: names = ["onepiece","onepiece(1)","onepiece(2)","onepiece(3)","onepiece"] Output: ["onepiece","onepiece(1)","onepiece(2)","onepiece(3)","onepiece(4)"] Explanation: When the last folder is created, the smallest positive valid k is 4, and it becomes "onepiece(4)".

Constraints:

1 <= names.length <= 5 * 104 1 <= names[i].length <= 20 names[i] consists of lowercase English letters, digits, and/or round brackets.

==
Problem Number: 1165 URL: https://leetcode.com/problems/avoid-flood-in-the-city Title: 1488. Avoid Flood in The City Problem Description: Your country has an infinite number of lakes. Initially, all the lakes are empty, but

when it rains over the nth lake, the nth lake becomes full of water. If it rains over a lake that is full of water, there will be a flood. Your goal is to avoid floods in any lake. Given an integer array rains where:

rains[i] > 0 means there will be rains over the rains[i] lake. rains[i] == 0 means there are no rains this day and you can choose one lake this day and dry it.

Return an array ans where:

ans.length == rains.length ans[i] == -1 if rains[i] > 0. ans[i] is the lake you choose to dry in the ith day if rains[i] == 0.

If there are multiple valid answers return any of them. If it is impossible to avoid flood return an empty array. Notice that if you chose to dry a full lake, it becomes empty, but if you chose to dry an empty lake, nothing changes. Example 1: Input: rains = [1,2,3,4] Output: [-1,-1,-1,-1] Explanation: After the first day full lakes are [1] After the second day full lakes are [1,2] After the third day full lakes are [1,2,3] After the fourth day full lakes are [1,2,3,4] There's no day to dry any lake and there is no flood in any lake.

Example 2: Input: rains = [1,2,0,0,2,1] Output: [-1,-1,2,1,-1,-1] Explanation: After the first day full lakes are [1] After the second day full lakes are [1,2] After the third day, we dry lake 2. Full lakes are [1] After the fourth day, we dry lake 1. There is no full lakes. After the fifth day, full lakes are [2]. After the sixth day, full lakes are [1,2]. It is easy that this scenario is flood-free. [-1,-1,1,2,-1,-1] is another acceptable scenario.

Example 3: Input: rains = [1,2,0,1,2] Output: [] Explanation: After the second day, full lakes are [1,2]. We have to dry one lake in the third day. After that, it will rain over lakes [1,2]. It's easy to prove that no matter which lake you choose to dry in the 3rd day, the other one will flood.

Constraints:

1 <= rains.length <= 105 0 <= rains[i] <= 109

==
Problem Number: 1166 URL: https://leetcode.com/problems/the-kth-factor-of-n Title: 1492. The kth Factor of n Problem Description: You are given two positive integers n and k. A factor of an integer n is defined as an integer i where n % i == 0. Consider a list of all factors of n sorted in ascending order, return the kth factor in this list or return -1 if n has less than k factors. Example 1: Input: n = 12, k = 3 Output: 3 Explanation: Factors list is [1, 2, 3, 4, 6, 12], the 3rd factor is 3.

Example 2: Input: n = 7, k = 2 Output: 7 Explanation: Factors list is [1, 7], the 2nd factor is 7.

Example 3: Input: n = 4, k = 4 Output: -1 Explanation: Factors list is [1, 2, 4], there is only 3 factors. We should return -1.

Constraints:

1 <= k <= n <= 1000

Follow up: Could you solve this problem in less than O(n) complexity?

==
Problem Number: 1167 URL: https://leetcode.com/problems/longest-subarray-of-1s-after-deleting-one-element Title: 1493. Longest Subarray of 1's After Deleting One Element Problem Description: Given a binary array nums, you should delete one element from it. Return the size of the longest non-empty subarray containing only 1's in the resulting array. Return 0 if there is no such subarray. Example 1: Input: nums = [1,1,0,1] Output: 3 Explanation: After deleting the number in position 2, [1,1,1] contains 3 numbers with value of 1's.

Example 2: Input: nums = [0,1,1,1,0,1,1,0,1] Output: 5 Explanation: After deleting the number in position 4, [0,1,1,1,1,1,0,1] longest subarray with value of 1's is [1,1,1,1,1].

Example 3: Input: nums = [1,1,1] Output: 2 Explanation: You must delete one element.

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1.

==
Problem Number: 1168 URL: https://leetcode.com/problems/check-if-array-pairs-are-divisible-by-k Title: 1497. Check If Array Pairs Are Divisible by k Problem Description: Given an array of integers arr of even length n and an integer k. We want to divide the array into exactly n / 2 pairs such that the sum of each pair is divisible by k. Return true If you can find a way to do that or false otherwise. Example 1: Input: arr = [1,2,3,4,5,10,6,7,8,9], k = 5 Output: true Explanation: Pairs are (1,9),(2,8),(3,7),(4,6) and (5,10).

Example 2: Input: arr = [1,2,3,4,5,6], k = 7 Output: true Explanation: Pairs are (1,6),(2,5) and(3,4).

Example 3: Input: arr = [1,2,3,4,5,6], k = 10 Output: false Explanation: You can try all possible pairs to see that there is no way to divide arr into 3 pairs each with sum divisible by 10.

Constraints:

arr.length == n 1 <= n <= 105 n is even. -109 <= arr[i] <= 109 1 <= k <= 105

==
Problem Number: 1169 URL: https://leetcode.com/problems/number-of-subsequences-that-satisfy-the-given-sum-condition Title: 1498. Number of Subsequences That Satisfy the Given Sum Condition Problem Description: You are given an array of integers nums and an integer target. Return the number of non-empty subsequences of nums such that the sum of the minimum and maximum element on it is less or equal to target. Since the answer may

be too large, return it modulo 109 + 7.   Example 1: Input: nums = [3,5,6,7], target = 9 Output: 4 Explanation: There are 4 subsequences that satisfy the condition. [3] -> Min value + max value <= target (3 + 3 <= 9) [3,5] -> (3 + 5 <= 9) [3,5,6] -> (3 + 6 <= 9) [3,6] -> (3 + 6 <= 9)

Example 2: Input: nums = [3,3,6,8], target = 10 Output: 6 Explanation: There are 6 subsequences that satisfy the condition. (nums can have repeated numbers). [3] , [3] , [3,3], [3,6] , [3,6] , [3,3,6]

Example 3: Input: nums = [2,3,3,4,6,7], target = 12 Output: 61 Explanation: There are 63 non-empty subsequences, two of them do not satisfy the condition ([6,7], [7]). Number of valid subsequences (63 - 2 = 61).

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 106 1 <= target <= 106

==
Problem Number: 1170 URL: https://leetcode.com/problems/last-moment-before-all-ants-fall-out-of-a-plank Title: 1503. Last Moment Before All Ants Fall Out of a Plank Problem Description: We have a wooden plank of the length n units. Some ants are walking on the plank, each ant moves with a speed of 1 unit per second. Some of the ants move to the left, the other move to the right. When two ants moving in two different directions meet at some point, they change their directions and continue moving again. Assume changing directions does not take any additional time. When an ant reaches one end of the plank at a time t, it falls out of the plank immediately. Given an integer n and two integer arrays left and right, the positions of the ants moving to the left and the right, return the moment when the last ant(s) fall out of the plank.   Example 1:

Input: n = 4, left = [4,3], right = [0,1] Output: 4 Explanation: In the image above: -The ant at index 0 is named A and going to the right. -The ant at index 1 is named B and going to the right. -The ant at index 3 is named C and going to the left. -The ant at index 4 is named D and going to the left. The last moment when an ant was on the plank is t = 4 seconds. After that, it falls immediately out of the plank. (i.e., We can say that at t = 4.0000000001, there are no ants on the plank).

Example 2:

Input: n = 7, left = [], right = [0,1,2,3,4,5,6,7] Output: 7 Explanation: All ants are going to the right, the ant at index 0 needs 7 seconds to fall.

Example 3:

Input: n = 7, left = [0,1,2,3,4,5,6,7], right = [] Output: 7 Explanation: All ants are going to the left, the ant at index 7 needs 7 seconds to fall.

Constraints:

1 <= n <= 104 0 <= left.length <= n + 1 0 <= left[i] <= n 0 <= right.length <= n + 1 0 <= right[i] <= n 1 <= left.length + right.length <= n + 1 All values of left and right are unique, and each value can appear only in one of the two arrays.

==
Problem Number: 1171 URL: https://leetcode.com/problems/count-submatrices-with-all-ones Title: 1504. Count Submatrices With All Ones Problem Description: Given an m x n binary matrix mat, return the number of submatrices that have all ones. Example 1:

Input: mat = [[1,0,1],[1,1,0],[1,1,0]] Output: 13 Explanation: There are 6 rectangles of side 1x1. There are 2 rectangles of side 1x2. There are 3 rectangles of side 2x1. There is 1 rectangle of side 2x2. There is 1 rectangle of side 3x1. Total number of rectangles = 6 + 2 + 3 + 1 + 1 = 13.

Example 2:

Input: mat = [[0,1,1,0],[0,1,1,1],[1,1,1,0]] Output: 24 Explanation: There are 8 rectangles of side 1x1. There are 5 rectangles of side 1x2. There are 2 rectangles of side 1x3. There are 4 rectangles of side 2x1. There are 2 rectangles of side 2x2. There are 2 rectangles of side 3x1. There is 1 rectangle of side 3x2. Total number of rectangles = 8 + 5 + 2 + 4 + 2 + 2 + 1 = 24.

Constraints:

1 <= m, n <= 150 mat[i][j] is either 0 or 1.

==
Problem Number: 1172 URL: https://leetcode.com/problems/range-sum-of-sorted-subarray-sums Title: 1508. Range Sum of Sorted Subarray Sums Problem Description: You are given the array nums consisting of n positive integers. You computed the sum of all non-empty continuous subarrays from the array and then sorted them in non-decreasing order, creating a new array of n * (n + 1) / 2 numbers. Return the sum of the numbers from index left to index right (indexed from 1), inclusive, in the new array. Since the answer can be a huge number return it modulo 109 + 7. Example 1: Input: nums = [1,2,3,4], n = 4, left = 1, right = 5 Output: 13 Explanation: All subarray sums are 1, 3, 6, 10, 2, 5, 9, 3, 7, 4. After sorting them in non-decreasing order we have the new array [1, 2, 3, 3, 4, 5, 6, 7, 9, 10]. The sum of the numbers from index le = 1 to ri = 5 is 1 + 2 + 3 + 3 + 4 = 13.

Example 2: Input: nums = [1,2,3,4], n = 4, left = 3, right = 4 Output: 6 Explanation: The given array is the same as example 1. We have the new array [1, 2, 3, 3, 4, 5, 6, 7, 9, 10]. The sum of the numbers from index le = 3 to ri = 4 is 3 + 3 = 6.

Example 3: Input: nums = [1,2,3,4], n = 4, left = 1, right = 10 Output: 50

Constraints:

n == nums.length 1 <= nums.length <= 1000 1 <= nums[i] <= 100 1 <= left <= right <= n * (n + 1) / 2

==
Problem Number: 1173 URL: https://leetcode.com/problems/minimum-difference-between-largest-and-smallest-value-in-three-moves Title: 1509. Minimum Difference Between Largest and Smallest Value in Three Moves Problem Description: You are given an integer array nums. In one move, you can choose one element of nums and change it to any value. Return the minimum difference between the largest and smallest value of nums after performing at most three moves. Example 1: Input: nums = [5,3,2,4] Output: 0 Explanation: We can make at most 3 moves. In the first move, change 2 to 3. nums becomes [5,3,3,4]. In the second move, change 4 to 3. nums becomes [5,3,3,3]. In the third move, change 5 to 3. nums becomes [3,3,3,3]. After performing 3 moves, the difference between the minimum and maximum is 3 - 3 = 0.

Example 2: Input: nums = [1,5,0,10,14] Output: 1 Explanation: We can make at most 3 moves. In the first move, change 5 to 0. nums becomes [1,0,0,10,14]. In the second move, change 10 to 0. nums becomes [1,0,0,0,14]. In the third move, change 14 to 1. nums becomes [1,0,0,0,1]. After performing 3 moves, the difference between the minimum and maximum is 1 - 0 = 1. It can be shown that there is no way to make the difference 0 in 3 moves. Example 3: Input: nums = [3,100,20] Output: 0 Explanation: We can make at most 3 moves. In the first move, change 100 to 7. nums becomes [3,7,20]. In the second move, change 20 to 7. nums becomes [3,7,7]. In the third move, change 3 to 7. nums becomes [7,7,7]. After performing 3 moves, the difference between the minimum and maximum is 7 - 7 = 0.

Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109

==
Problem Number: 1174 URL: https://leetcode.com/problems/number-of-substrings-with-only-1s Title: 1513. Number of Substrings With Only 1s Problem Description: Given a binary string s, return the number of substrings with all characters 1's. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: s = "0110111" Output: 9 Explanation: There are 9 substring in total with only 1's characters. "1" -> 5 times. "11" -> 3 times. "111" -> 1 time. Example 2: Input: s = "101" Output: 2 Explanation: Substring "1" is shown 2 times in s.

Example 3: Input: s = "111111" Output: 21 Explanation: Each substring contains only 1's characters.

Constraints:

1 <= s.length <= 105 s[i] is either '0' or '1'.

== Problem Number: 1175 URL: https://leetcode.com/problems/path-with-maximum-probability Title: 1514. Path with Maximum Probability Problem Description: You are given an undirected weighted graph of n nodes (0-indexed), represented by an edge list where edges[i] = [a, b] is an undirected edge connecting the nodes a and b with a probability of success of traversing that edge succProb[i]. Given two nodes start and end, find the path with the maximum probability of success to go from start to end and return its success probability. If there is no path from start to end, return 0. Your answer will be accepted if it differs from the correct answer by at most 1e-5. Example 1:

Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.2], start = 0, end = 2 Output: 0.25000 Explanation: There are two paths from start to end, one having a probability of success = 0.2 and the other has 0.5 * 0.5 = 0.25.

Example 2:

Input: n = 3, edges = [[0,1],[1,2],[0,2]], succProb = [0.5,0.5,0.3], start = 0, end = 2 Output: 0.30000

Example 3:

Input: n = 3, edges = [[0,1]], succProb = [0.5], start = 0, end = 2 Output: 0.00000 Explanation: There is no path between 0 and 2.

Constraints:

2 <= n <= 10^4 0 <= start, end < n start != end 0 <= a, b < n a != b 0 <= succProb.length == edges.length <= 2*10^4 0 <= succProb[i] <= 1 There is at most one edge between every two nodes.

== Problem Number: 1176 URL: https://leetcode.com/problems/number-of-nodes-in-the-sub-tree-with-the-same-label Title: 1519. Number of Nodes in the Sub-Tree With the Same Label Problem Description: You are given a tree (i.e. a connected, undirected graph that has no cycles) consisting of n nodes numbered from 0 to n - 1 and exactly n - 1 edges. The root of the tree is the node 0, and each node of the tree has a label which is a lower-case character given in the string labels (i.e. The node with the number i has the label labels[i]). The edges array is given on the form edges[i] = [ai, bi], which means there is an edge between nodes ai and bi in the tree. Return an array of size n where ans[i] is the number of nodes in the subtree of the ith node which have the same label as node i. A subtree of a tree T is the tree consisting of a node in T and all of its descendant nodes. Example 1:

Input: n = 7, edges = [[0,1],[0,2],[1,4],[1,5],[2,3],[2,6]], labels = "abaedcd" Output: [2,1,1,1,1,1,1] Explanation: Node 0 has label 'a' and its sub-tree has node 2 with label 'a' as well, thus the answer is 2. Notice that any node is part of its sub-tree. Node 1 has a label 'b'. The sub-tree of node 1 contains nodes 1,4 and

5, as nodes 4 and 5 have different labels than node 1, the answer is just 1 (the node itself).

Example 2:

Input: n = 4, edges = [[0,1],[1,2],[0,3]], labels = "bbbb" Output: [4,2,1,1] Explanation: The sub-tree of node 2 contains only node 2, so the answer is 1. The sub-tree of node 3 contains only node 3, so the answer is 1. The sub-tree of node 1 contains nodes 1 and 2, both have label 'b', thus the answer is 2. The sub-tree of node 0 contains nodes 0, 1, 2 and 3, all with label 'b', thus the answer is 4.

Example 3:

Input: n = 5, edges = [[0,1],[0,2],[1,3],[0,4]], labels = "aabab" Output: [3,2,1,1,1]

Constraints:

1 <= n <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi labels.length == n labels is consisting of only of lowercase English letters.

==
Problem Number: 1177 URL: https://leetcode.com/problems/number-of-sub-arrays-with-odd-sum Title: 1524. Number of Sub-arrays With Odd Sum Problem Description: Given an array of integers arr, return the number of subarrays with an odd sum. Since the answer can be very large, return it modulo 109 + 7. Example 1: Input: arr = [1,3,5] Output: 4 Explanation: All subarrays are [[1],[1,3],[1,3,5],[3],[3,5],[5]] All sub-arrays sum are [1,4,9,3,8,5]. Odd sums are [1,9,3,5] so the answer is 4.

Example 2: Input: arr = [2,4,6] Output: 0 Explanation: All subarrays are [[2],[2,4],[2,4,6],[4],[4,6],[6]] All sub-arrays sum are [2,6,12,4,10,6]. All sub-arrays have even sum and the answer is 0.

Example 3: Input: arr = [1,2,3,4,5,6,7] Output: 16

Constraints:

1 <= arr.length <= 105 1 <= arr[i] <= 100

==
Problem Number: 1178 URL: https://leetcode.com/problems/number-of-good-ways-to-split-a-string Title: 1525. Number of Good Ways to Split a String Problem Description: You are given a string s. A split is called good if you can split s into two non-empty strings sleft and sright where their concatenation is equal to s (i.e., sleft + sright = s) and the number of distinct letters in sleft and sright is the same. Return the number of good splits you can make in s. Example 1: Input: s = "aacaba" Output: 2 Explanation: There are 5 ways to split "aacaba" and 2 of them are good. ("a", "acaba") Left string and right string contains 1 and 3 different letters respectively. ("aa", "caba") Left string and right string contains 1 and 3 different letters respectively. ("aac", "aba") Left string and right string contains 2 and 2 different letters respectively (good split). ("aaca", "ba") Left string and right string contains 2 and 2 different

letters respectively (good split). ("aacab", "a") Left string and right string contains 3 and 1 different letters respectively.

Example 2: Input: s = "abcd" Output: 1 Explanation: Split the string as follows ("ab", "cd").

Constraints:

1 <= s.length <= 105 s consists of only lowercase English letters.

==

Problem Number: 1179 URL: https://leetcode.com/problems/minimum-suffix-flips Title: 1529. Minimum Suffix Flips Problem Description: You are given a 0-indexed binary string target of length n. You have another binary string s of length n that is initially set to all zeros. You want to make s equal to target. In one operation, you can pick an index i where 0 <= i < n and flip all bits in the inclusive range [i, n - 1]. Flip means changing '0' to '1' and '1' to '0'. Return the minimum number of operations needed to make s equal to target. Example 1: Input: target = "10111" Output: 3 Explanation: Initially, s = "00000". Choose index i = 2: "00000" -> "00111" Choose index i = 0: "00111" -> "11000" Choose index i = 1: "11000" -> "10111" We need at least 3 flip operations to form target.

Example 2: Input: target = "101" Output: 3 Explanation: Initially, s = "000". Choose index i = 0: "000" -> "111" Choose index i = 1: "111" -> "100" Choose index i = 2: "100" -> "101" We need at least 3 flip operations to form target.

Example 3: Input: target = "00000" Output: 0 Explanation: We do not need any operations since the initial s already equals target.

Constraints:

n == target.length 1 <= n <= 105 target[i] is either '0' or '1'.

==

Problem Number: 1180 URL: https://leetcode.com/problems/number-of-good-leaf-nodes-pairs Title: 1530. Number of Good Leaf Nodes Pairs Problem Description: You are given the root of a binary tree and an integer distance. A pair of two different leaf nodes of a binary tree is said to be good if the length of the shortest path between them is less than or equal to distance. Return the number of good leaf node pairs in the tree. Example 1:

Input: root = [1,2,3,null,4], distance = 3 Output: 1 Explanation: The leaf nodes of the tree are 3 and 4 and the length of the shortest path between them is 3. This is the only good pair.

Example 2:

Input: root = [1,2,3,4,5,6,7], distance = 3 Output: 2 Explanation: The good pairs are [4,5] and [6,7] with shortest path = 2. The pair [4,6] is not good because the length of ther shortest path between them is 4.

Example 3: Input: root = [7,1,4,6,null,5,3,null,null,null,null,null,2], distance = 3 Output: 1 Explanation: The only good pair is [2,5].

Constraints:

The number of nodes in the tree is in the range [1, 210]. 1 <= Node.val <= 100 1 <= distance <= 10

==
Problem Number: 1181 URL: https://leetcode.com/problems/find-the-winner-of-an-array-game Title: 1535. Find the Winner of an Array Game Problem Description: Given an integer array arr of distinct integers and an integer k. A game will be played between the first two elements of the array (i.e. arr[0] and arr[1]). In each round of the game, we compare arr[0] with arr[1], the larger integer wins and remains at position 0, and the smaller integer moves to the end of the array. The game ends when an integer wins k consecutive rounds. Return the integer which will win the game. It is guaranteed that there will be a winner of the game.   Example 1: Input: arr = [2,1,3,5,4,6,7], k = 2 Output: 5 Explanation: Let's see the rounds of the game: Round | arr | winner | win_count 1 | [2,1,3,5,4,6,7] | 2 | 1 2 | [2,3,5,4,6,7,1] | 3 | 1 3 | [3,5,4,6,7,1,2] | 5 | 1 4 | [5,4,6,7,1,2,3] | 5 | 2 So we can see that 4 rounds will be played and 5 is the winner because it wins 2 consecutive games.

Example 2: Input: arr = [3,2,1], k = 10 Output: 3 Explanation: 3 will win the first 10 rounds consecutively.

Constraints:

2 <= arr.length <= 105 1 <= arr[i] <= 106 arr contains distinct integers. 1 <= k <= 109

==
Problem Number: 1182 URL: https://leetcode.com/problems/minimum-swaps-to-arrange-a-binary-grid Title: 1536. Minimum Swaps to Arrange a Binary Grid Problem Description: Given an n x n binary grid, in one step you can choose two adjacent rows of the grid and swap them. A grid is said to be valid if all the cells above the main diagonal are zeros. Return the minimum number of steps needed to make the grid valid, or -1 if the grid cannot be valid. The main diagonal of a grid is the diagonal that starts at cell (1, 1) and ends at cell (n, n).   Example 1:

Input: grid = [[0,0,1],[1,1,0],[1,0,0]] Output: 3

Example 2:

Input: grid = [[0,1,1,0],[0,1,1,0],[0,1,1,0],[0,1,1,0]] Output: -1 Explanation: All rows are similar, swaps have no effect on the grid.

Example 3:

Input: grid = [[1,0,0],[1,1,0],[1,1,1]] Output: 0

Constraints:

n == grid.length == grid[i].length 1 <= n <= 200 grid[i][j] is either 0 or 1

==
Problem Number: 1183 URL: https://leetcode.com/problems/can-convert-string-in-k-moves Title: 1540. Can Convert String in K Moves Problem Description: Given two strings s and t, your goal is to convert s into t in k moves or less. During the ith (1 <= i <= k) move you can:

Choose any index j (1-indexed) from s, such that 1 <= j <= s.length and j has not been chosen in any previous move, and shift the character at that index i times. Do nothing.

Shifting a character means replacing it by the next letter in the alphabet (wrapping around so that 'z' becomes 'a'). Shifting a character by i means applying the shift operations i times. Remember that any index j can be picked at most once. Return true if it's possible to convert s into t in no more than k moves, otherwise return false. Example 1: Input: s = "input", t = "ouput", k = 9 Output: true Explanation: In the 6th move, we shift 'i' 6 times to get 'o'. And in the 7th move we shift 'n' to get 'u'.

Example 2: Input: s = "abc", t = "bcd", k = 10 Output: false Explanation: We need to shift each character in s one time to convert it into t. We can shift 'a' to 'b' during the 1st move. However, there is no way to shift the other characters in the remaining moves to obtain t from s.

Example 3: Input: s = "aab", t = "bbb", k = 27 Output: true Explanation: In the 1st move, we shift the first 'a' 1 time to get 'b'. In the 27th move, we shift the second 'a' 27 times to get 'b'.

Constraints:

1 <= s.length, t.length <= 10^5 0 <= k <= 10^9 s, t contain only lowercase English letters.

==
Problem Number: 1184 URL: https://leetcode.com/problems/minimum-insertions-to-balance-a-parentheses-string Title: 1541. Minimum Insertions to Balance a Parentheses String Problem Description: Given a parentheses string s containing only the characters '(' and ')'. A parentheses string is balanced if:

Any left parenthesis '(' must have a corresponding two consecutive right parenthesis '))'. Left parenthesis '(' must go before the corresponding two consecutive right parenthesis '))'.

In other words, we treat '(' as an opening parenthesis and '))' as a closing parenthesis.

For example, "())", "())(())))" and "(())())))" are balanced, ")()", "()))" and "(()))" are not balanced.

You can insert the characters '(' and ')' at any position of the string to balance it if needed. Return the minimum number of insertions needed to make s balanced. Example 1: Input: s = "(()))" Output: 1 Explanation: The second '(' has two matching '))', but the first '(' has only ')' matching. We need to add one more ')' at the end of the string to be "(())))" which is balanced.

Example 2: Input: s = "())" Output: 0 Explanation: The string is already balanced.

Example 3: Input: s = "))())(" Output: 3 Explanation: Add '(' to match the first '))', Add '))' to match the last '('.

Constraints:

1 <= s.length <= 105 s consists of '(' and ')' only.

==
Problem Number: 1185 URL: https://leetcode.com/problems/find-kth-bit-in-nth-binary-string Title: 1545. Find Kth Bit in Nth Binary String Problem Description: Given two positive integers n and k, the binary string Sn is formed as follows:

S1 = "0" Si = Si - 1 + "1" + reverse(invert(Si - 1)) for i > 1

Where + denotes the concatenation operation, reverse(x) returns the reversed string x, and invert(x) inverts all the bits in x (0 changes to 1 and 1 changes to 0). For example, the first four strings in the above sequence are:

S1 = "0" S2 = "011" S3 = "0111001" S4 = "011100110110001"

Return the kth bit in Sn. It is guaranteed that k is valid for the given n. Example 1: Input: n = 3, k = 1 Output: "0" Explanation: S3 is "0111001". The 1st bit is "0".

Example 2: Input: n = 4, k = 11 Output: "1" Explanation: S4 is "011100110110001". The 11th bit is "1".

Constraints:

1 <= n <= 20 1 <= k <= 2n - 1

==
Problem Number: 1186 URL: https://leetcode.com/problems/maximum-number-of-non-overlapping-subarrays-with-sum-equals-target Title: 1546. Maximum Number of Non-Overlapping Subarrays With Sum Equals Target Problem Description: Given an array nums and an integer target, return the maximum number of non-empty non-overlapping subarrays such that the sum of values in each subarray is equal to target. Example 1: Input: nums = [1,1,1,1,1], target = 2 Output: 2 Explanation: There are 2 non-overlapping subarrays [1,1,1,1,1] with sum equals to target(2).

Example 2: Input: nums = [-1,3,5,1,4,2,-9], target = 6 Output: 2 Explanation: There are 3 subarrays with sum equal to 6. ([5,1], [4,2], [3,5,1,4,2,-9]) but only the first 2 are non-overlapping.

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104 0 <= target <= 106

==
Problem Number: 1187 URL: https://leetcode.com/problems/minimum-operations-to-make-array-equal Title: 1551. Minimum Operations to Make Array Equal Problem Description: You have an array arr of length n where arr[i] = (2 * i) + 1 for all valid values of i (i.e., 0 <= i < n). In one operation, you can select two indices x and y where 0 <= x, y < n and subtract 1 from arr[x] and add 1 to arr[y] (i.e., perform arr[x] -=1 and arr[y] += 1). The goal is to make all the elements of the array equal. It is guaranteed that all the elements of the array can be made equal using some operations. Given an integer n, the length of the array, return the minimum number of operations needed to make all the elements of arr equal. Example 1: Input: n = 3 Output: 2 Explanation: arr = [1, 3, 5] First operation choose x = 2 and y = 0, this leads arr to be [2, 3, 4] In the second operation choose x = 2 and y = 0 again, thus arr = [3, 3, 3].

Example 2: Input: n = 6 Output: 9

Constraints:

1 <= n <= 104

==
Problem Number: 1188 URL: https://leetcode.com/problems/magnetic-force-between-two-balls Title: 1552. Magnetic Force Between Two Balls Problem Description: In the universe Earth C-137, Rick discovered a special form of magnetic force between two balls if they are put in his new invented basket. Rick has n empty baskets, the ith basket is at position[i], Morty has m balls and needs to distribute the balls into the baskets such that the minimum magnetic force between any two balls is maximum. Rick stated that magnetic force between two different balls at positions x and y is |x - y|. Given the integer array position and the integer m. Return the required force. Example 1:

Input: position = [1,2,3,4,7], m = 3 Output: 3 Explanation: Distributing the 3 balls into baskets 1, 4 and 7 will make the magnetic force between ball pairs [3, 3, 6]. The minimum magnetic force is 3. We cannot achieve a larger minimum magnetic force than 3.

Example 2: Input: position = [5,4,3,2,1,1000000000], m = 2 Output: 999999999 Explanation: We can use baskets 1 and 1000000000.

Constraints:

n == position.length 2 <= n <= 105 1 <= position[i] <= 109 All integers in position are distinct. 2 <= m <= position.length

==

Problem Number: 1189 URL: https://leetcode.com/problems/minimum-number-of-vertices-to-reach-all-nodes Title: 1557. Minimum Number of Vertices to Reach All Nodes Problem Description: Given a directed acyclic graph, with n vertices numbered from 0 to n-1, and an array edges where edges[i] = [fromi, toi] represents a directed edge from node fromi to node toi. Find the smallest set of vertices from which all nodes in the graph are reachable. It's guaranteed that a unique solution exists. Notice that you can return the vertices in any order. Example 1:

Input: n = 6, edges = [[0,1],[0,2],[2,5],[3,4],[4,2]] Output: [0,3] Explanation: It's not possible to reach all the nodes from a single vertex. From 0 we can reach [0,1,2,5]. From 3 we can reach [3,4,2,5]. So we output [0,3]. Example 2:

Input: n = 5, edges = [[0,1],[2,1],[3,1],[1,4],[2,4]] Output: [0,2,3] Explanation: Notice that vertices 0, 3 and 2 are not reachable from any other node, so we must include them. Also any of these vertices can reach nodes 1 and 4.

Constraints:

2 <= n <= 10^5 1 <= edges.length <= min(10^5, n * (n - 1) / 2) edges[i].length == 2 0 <= fromi, toi < n All pairs (fromi, toi) are distinct.

==

Problem Number: 1190 URL: https://leetcode.com/problems/minimum-numbers-of-function-calls-to-make-target-array Title: 1558. Minimum Numbers of Function Calls to Make Target Array Problem Description: You are given an integer array nums. You have an integer array arr of the same length with all values set to 0 initially. You also have the following modify function:

You want to use the modify function to convert arr to nums using the minimum number of calls. Return the minimum number of function calls to make nums from arr. The test cases are generated so that the answer fits in a 32-bit signed integer. Example 1: Input: nums = [1,5] Output: 5 Explanation: Increment by 1 (second element): [0, 0] to get [0, 1] (1 operation). Double all the elements: [0, 1] -> [0, 2] -> [0, 4] (2 operations). Increment by 1 (both elements) [0, 4] -> [1, 4] -> [1, 5] (2 operations). Total of operations: 1 + 2 + 2 = 5.

Example 2: Input: nums = [2,2] Output: 3 Explanation: Increment by 1 (both elements) [0, 0] -> [0, 1] -> [1, 1] (2 operations). Double all the elements: [1, 1] -> [2, 2] (1 operation). Total of operations: 2 + 1 = 3.

Example 3: Input: nums = [4,2,5] Output: 6 Explanation: (initial)[0,0,0] -> [1,0,0] -> [1,0,1] -> [2,0,2] -> [2,1,2] -> [4,2,4] -> [4,2,5](nums).

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==

Problem Number: 1191 URL: https://leetcode.com/problems/detect-cycles-in-

2d-grid Title: 1559. Detect Cycles in 2D Grid Problem Description: Given a 2D array of characters grid of size m x n, you need to find if there exists any cycle consisting of the same value in grid. A cycle is a path of length 4 or more in the grid that starts and ends at the same cell. From a given cell, you can move to one of the cells adjacent to it - in one of the four directions (up, down, left, or right), if it has the same value of the current cell. Also, you cannot move to the cell that you visited in your last move. For example, the cycle (1, 1) -> (1, 2) -> (1, 1) is invalid because from (1, 2) we visited (1, 1) which was the last visited cell. Return true if any cycle of the same value exists in grid, otherwise, return false. Example 1:

Input: grid = [["a","a","a","a"],["a","b","b","a"],["a","b","b","a"],["a","a","a","a"]] Output: true Explanation: There are two valid cycles shown in different colors in the image below:

Example 2:

Input: grid = [["c","c","c","a"],["c","d","c","c"],["c","c","e","c"],["f","c","c","c"]] Output: true Explanation: There is only one valid cycle highlighted in the image below:

Example 3:

Input: grid = [["a","b","b"],["b","z","b"],["b","b","a"]] Output: false

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 500 grid consists only of lowercase English letters.

===

Problem Number: 1192 URL: https://leetcode.com/problems/maximum-number-of-coins-you-can-get Title: 1561. Maximum Number of Coins You Can Get Problem Description: There are 3n piles of coins of varying size, you and your friends will take piles of coins as follows:

In each step, you will choose any 3 piles of coins (not necessarily consecutive). Of your choice, Alice will pick the pile with the maximum number of coins. You will pick the next pile with the maximum number of coins. Your friend Bob will pick the last pile. Repeat until there are no more piles of coins.

Given an array of integers piles where piles[i] is the number of coins in the ith pile. Return the maximum number of coins that you can have. Example 1: Input: piles = [2,4,1,2,7,8] Output: 9 Explanation: Choose the triplet (2, 7, 8), Alice Pick the pile with 8 coins, you the pile with 7 coins and Bob the last one. Choose the triplet (1, 2, 4), Alice Pick the pile with 4 coins, you the pile with 2 coins and Bob the last one. The maximum number of coins which you can have are: 7 + 2 = 9. On the other hand if we choose this arrangement (1, 2, 8), (2, 4, 7) you only get 2 + 4 = 6 coins which is not optimal.

Example 2: Input: piles = [2,4,5] Output: 4

Example 3: Input: piles = [9,8,7,6,5,1,2,3,4] Output: 18

Constraints:

3 <= piles.length <= 105 piles.length % 3 == 0 1 <= piles[i] <= 104

==

Problem Number: 1193 URL: https://leetcode.com/problems/find-latest-group-of-size-m Title: 1562. Find Latest Group of Size M Problem Description: Given an array arr that represents a permutation of numbers from 1 to n. You have a binary string of size n that initially has all its bits set to zero. At each step i (assuming both the binary string and arr are 1-indexed) from 1 to n, the bit at position arr[i] is set to 1. You are also given an integer m. Find the latest step at which there exists a group of ones of length m. A group of ones is a contiguous substring of 1’s such that it cannot be extended in either direction. Return the latest step at which there exists a group of ones of length exactly m. If no such group exists, return -1. Example 1: Input: arr = [3,5,1,2,4], m = 1 Output: 4 Explanation: Step 1: ”00100”, groups: [”1”] Step 2: ”00101”, groups: [”1”, ”1”] Step 3: ”10101”, groups: [”1”, ”1”, ”1”] Step 4: ”11101”, groups: [”111”, ”1”] Step 5: ”11111”, groups: [”11111”] The latest step at which there exists a group of size 1 is step 4.

Example 2: Input: arr = [3,1,5,4,2], m = 2 Output: -1 Explanation: Step 1: ”00100”, groups: [”1”] Step 2: ”10100”, groups: [”1”, ”1”] Step 3: ”10101”, groups: [”1”, ”1”, ”1”] Step 4: ”10111”, groups: [”1”, ”111”] Step 5: ”11111”, groups: [”11111”] No group of size 2 exists during any step.

Constraints:

n == arr.length 1 <= m <= n <= 105 1 <= arr[i] <= n All integers in arr are distinct.

==

Problem Number: 1194 URL: https://leetcode.com/problems/maximum-length-of-subarray-with-positive-product Title: 1567. Maximum Length of Subarray With Positive Product Problem Description: Given an array of integers nums, find the maximum length of a subarray where the product of all its elements is positive. A subarray of an array is a consecutive sequence of zero or more values taken out of that array. Return the maximum length of a subarray with positive product. Example 1: Input: nums = [1,-2,-3,4] Output: 4 Explanation: The array nums already has a positive product of 24.

Example 2: Input: nums = [0,1,-2,-3,-4] Output: 3 Explanation: The longest subarray with positive product is [1,-2,-3] which has a product of 6. Notice that we cannot include 0 in the subarray since that’ll make the product 0 which is not positive. Example 3: Input: nums = [-1,-2,-3,0,1] Output: 2 Explanation: The longest subarray with positive product is [-1,-2] or [-2,-3].

Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109

==
Problem Number: 1195 URL: https://leetcode.com/problems/number-of-ways-to-split-a-string Title: 1573. Number of Ways to Split a String Problem Description: Given a binary string s, you can split s into 3 non-empty strings s1, s2, and s3 where s1 + s2 + s3 = s. Return the number of ways s can be split such that the number of ones is the same in s1, s2, and s3. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: s = ”10101” Output: 4 Explanation: There are four ways to split s in 3 parts where each part contain the same number of letters ’1’. ”1|010|1” ”1|01|01” ”10|10|1” ”10|1|01”

Example 2: Input: s = ”1001” Output: 0

Example 3: Input: s = ”0000” Output: 3 Explanation: There are three ways to split s in 3 parts. ”0|0|00” ”0|00|0” ”00|0|0”

Constraints:

3 <= s.length <= 105 s[i] is either ’0’ or ’1’.

==
Problem Number: 1196 URL: https://leetcode.com/problems/shortest-subarray-to-be-removed-to-make-array-sorted Title: 1574. Shortest Subarray to be Removed to Make Array Sorted Problem Description: Given an integer array arr, remove a subarray (can be empty) from arr such that the remaining elements in arr are non-decreasing. Return the length of the shortest subarray to remove. A subarray is a contiguous subsequence of the array. Example 1: Input: arr = [1,2,3,10,4,2,3,5] Output: 3 Explanation: The shortest subarray we can remove is [10,4,2] of length 3. The remaining elements after that will be [1,2,3,3,5] which are sorted. Another correct solution is to remove the subarray [3,10,4].

Example 2: Input: arr = [5,4,3,2,1] Output: 4 Explanation: Since the array is strictly decreasing, we can only keep a single element. Therefore we need to remove a subarray of length 4, either [5,4,3,2] or [4,3,2,1].

Example 3: Input: arr = [1,2,3] Output: 0 Explanation: The array is already non-decreasing. We do not need to remove any elements.

Constraints:

1 <= arr.length <= 105 0 <= arr[i] <= 109

==
Problem Number: 1197 URL: https://leetcode.com/problems/number-of-ways-where-square-of-number-is-equal-to-product-of-two-numbers Title: 1577. Number of Ways Where Square of Number Is Equal to Product of Two Numbers Problem Description: Given two arrays of integers nums1 and nums2, return the number of triplets formed (type 1 and type 2) under the following rules:

Type 1: Triplet (i, j, k) if nums1[i]2 == nums2[j] * nums2[k] where 0 <= i < nums1.length and 0 <= j < k < nums2.length. Type 2: Triplet (i, j, k) if nums2[i]2 == nums1[j] * nums1[k] where 0 <= i < nums2.length and 0 <= j < k < nums1.length.

Example 1: Input: nums1 = [7,4], nums2 = [5,2,8,9] Output: 1 Explanation: Type 1: (1, 1, 2), nums1[1]2 = nums2[1] * nums2[2]. (42 = 2 * 8).

Example 2: Input: nums1 = [1,1], nums2 = [1,1,1] Output: 9 Explanation: All Triplets are valid, because 12 = 1 * 1. Type 1: (0,0,1), (0,0,2), (0,1,2), (1,0,1), (1,0,2), (1,1,2). nums1[i]2 = nums2[j] * nums2[k]. Type 2: (0,0,1), (1,0,1), (2,0,1). nums2[i]2 = nums1[j] * nums1[k].

Example 3: Input: nums1 = [7,7,8,3], nums2 = [1,2,9,7] Output: 2 Explanation: There are 2 valid triplets. Type 1: (3,0,2). nums1[3]2 = nums2[0] * nums2[2]. Type 2: (3,0,1). nums2[3]2 = nums1[0] * nums1[1].

Constraints:

1 <= nums1.length, nums2.length <= 1000 1 <= nums1[i], nums2[i] <= 105

==

Problem Number: 1198 URL: https://leetcode.com/problems/minimum-time-to-make-rope-colorful Title: 1578. Minimum Time to Make Rope Colorful Problem Description: Alice has n balloons arranged on a rope. You are given a 0-indexed string colors where colors[i] is the color of the ith balloon. Alice wants the rope to be colorful. She does not want two consecutive balloons to be of the same color, so she asks Bob for help. Bob can remove some balloons from the rope to make it colorful. You are given a 0-indexed integer array neededTime where neededTime[i] is the time (in seconds) that Bob needs to remove the ith balloon from the rope. Return the minimum time Bob needs to make the rope colorful. Example 1:

Input: colors = "abaac", neededTime = [1,2,3,4,5] Output: 3 Explanation: In the above image, 'a' is blue, 'b' is red, and 'c' is green. Bob can remove the blue balloon at index 2. This takes 3 seconds. There are no longer two consecutive balloons of the same color. Total time = 3. Example 2:

Input: colors = "abc", neededTime = [1,2,3] Output: 0 Explanation: The rope is already colorful. Bob does not need to remove any balloons from the rope.

Example 3:

Input: colors = "aabaa", neededTime = [1,2,3,4,1] Output: 2 Explanation: Bob will remove the ballons at indices 0 and 4. Each ballon takes 1 second to remove. There are no longer two consecutive balloons of the same color. Total time = 1 + 1 = 2.

Constraints:

n == colors.length == neededTime.length 1 <= n <= 105 1 <= neededTime[i] <= 104 colors contains only lowercase English letters.

==

Problem Number: 1199 URL: https://leetcode.com/problems/count-unhappy-friends Title: 1583. Count Unhappy Friends Problem Description: You are given a list of preferences for n friends, where n is always even. For each person i, preferences[i] contains a list of friends sorted in the order of preference. In other words, a friend earlier in the list is more preferred than a friend later in the list. Friends in each list are denoted by integers from 0 to n-1. All the friends are divided into pairs. The pairings are given in a list pairs, where pairs[i] = [xi, yi] denotes xi is paired with yi and yi is paired with xi. However, this pairing may cause some of the friends to be unhappy. A friend x is unhappy if x is paired with y and there exists a friend u who is paired with v but:

x prefers u over y, and u prefers x over v.

Return the number of unhappy friends.   Example 1: Input: n = 4, preferences = [[1, 2, 3], [3, 2, 0], [3, 1, 0], [1, 2, 0]], pairs = [[0, 1], [2, 3]] Output: 2 Explanation: Friend 1 is unhappy because: - 1 is paired with 0 but prefers 3 over 0, and - 3 prefers 1 over 2. Friend 3 is unhappy because: - 3 is paired with 2 but prefers 1 over 2, and - 1 prefers 3 over 0. Friends 0 and 2 are happy.

Example 2: Input: n = 2, preferences = [[1], [0]], pairs = [[1, 0]] Output: 0 Explanation: Both friends 0 and 1 are happy.

Example 3: Input: n = 4, preferences = [[1, 3, 2], [2, 3, 0], [1, 3, 0], [0, 2, 1]], pairs = [[1, 3], [0, 2]] Output: 4

Constraints:

2 <= n <= 500 n is even. preferences.length == n preferences[i].length == n - 1 0 <= preferences[i][j] <= n - 1 preferences[i] does not contain i. All values in preferences[i] are unique. pairs.length == n/2 pairs[i].length == 2 xi != yi 0 <= xi, yi <= n - 1 Each person is contained in exactly one pair.

==

Problem Number: 1200 URL: https://leetcode.com/problems/min-cost-to-connect-all-points Title: 1584. Min Cost to Connect All Points Problem Description: You are given an array points representing integer coordinates of some points on a 2D-plane, where points[i] = [xi, yi]. The cost of connecting two points [xi, yi] and [xj, yj] is the manhattan distance between them: |xi - xj| + |yi - yj|, where |val| denotes the absolute value of val. Return the minimum cost to make all points connected. All points are connected if there is exactly one simple path between any two points.   Example 1:

Input: points = [[0,0],[2,2],[3,10],[5,2],[7,0]] Output: 20 Explanation:

We can connect the points as shown above to get the minimum cost of 20. Notice that there is a unique path between every pair of points.

Example 2: Input: points = [[3,12],[-2,5],[-4,1]] Output: 18

Constraints:

1 <= points.length <= 1000 -106 <= xi, yi <= 106 All pairs (xi, yi) are distinct.

==
Problem Number: 1201 URL: https://leetcode.com/problems/maximum-sum-obtained-of-any-permutation Title: 1589. Maximum Sum Obtained of Any Permutation Problem Description: We have an array of integers, nums, and an array of requests where requests[i] = [starti, endi]. The ith request asks for the sum of nums[starti] + nums[starti + 1] + ... + nums[endi - 1] + nums[endi]. Both starti and endi are 0-indexed. Return the maximum total sum of all requests among all permutations of nums. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: nums = [1,2,3,4,5], requests = [[1,3],[0,1]] Output: 19 Explanation: One permutation of nums is [2,1,3,4,5] with the following result: requests[0] -> nums[1] + nums[2] + nums[3] = 1 + 3 + 4 = 8 requests[1] -> nums[0] + nums[1] = 2 + 1 = 3 Total sum: 8 + 3 = 11. A permutation with a higher total sum is [3,5,4,2,1] with the following result: requests[0] -> nums[1] + nums[2] + nums[3] = 5 + 4 + 2 = 11 requests[1] -> nums[0] + nums[1] = 3 + 5 = 8 Total sum: 11 + 8 = 19, which is the best that you can do.

Example 2: Input: nums = [1,2,3,4,5,6], requests = [[0,1]] Output: 11 Explanation: A permutation with the max total sum is [6,5,4,3,2,1] with request sums [11]. Example 3: Input: nums = [1,2,3,4,5,10], requests = [[0,2],[1,3],[1,1]] Output: 47 Explanation: A permutation with the max total sum is [4,10,5,3,2,1] with request sums [19,18,10]. Constraints:

n == nums.length 1 <= n <= 105 0 <= nums[i] <= 105 1 <= requests.length <= 105 requests[i].length == 2 0 <= starti <= endi < n

==
Problem Number: 1202 URL: https://leetcode.com/problems/make-sum-divisible-by-p Title: 1590. Make Sum Divisible by P Problem Description: Given an array of positive integers nums, remove the smallest subarray (possibly empty) such that the sum of the remaining elements is divisible by p. It is not allowed to remove the whole array. Return the length of the smallest subarray that you need to remove, or -1 if it's impossible. A subarray is defined as a contiguous block of elements in the array. Example 1: Input: nums = [3,1,4,2], p = 6 Output: 1 Explanation: The sum of the elements in nums is 10, which is not divisible by 6. We can remove the subarray [4], and the sum of the remaining elements is 6, which is divisible by 6.

Example 2: Input: nums = [6,3,5,2], p = 9 Output: 2 Explanation: We cannot remove a single element to get a sum divisible by 9. The best way is to remove the subarray [5,2], leaving us with [6,3] with sum 9.

Example 3: Input: nums = [1,2,3], p = 3 Output: 0 Explanation: Here the sum is 6. which is already divisible by 3. Thus we do not need to remove anything.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= p <= 109

==
Problem Number: 1203 URL: https://leetcode.com/problems/split-a-string-into-the-max-number-of-unique-substrings Title: 1593. Split a String Into the Max Number of Unique Substrings Problem Description: Given a string s, return the maximum number of unique substrings that the given string can be split into. You can split string s into any list of non-empty substrings, where the concatenation of the substrings forms the original string. However, you must split the substrings such that all of them are unique. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "ababccc" Output: 5 Explanation: One way to split maximally is ['a', 'b', 'ab', 'c', 'cc']. Splitting like ['a', 'b', 'a', 'b', 'c', 'cc'] is not valid as you have 'a' and 'b' multiple times.

Example 2: Input: s = "aba" Output: 2 Explanation: One way to split maximally is ['a', 'ba'].

Example 3: Input: s = "aa" Output: 1 Explanation: It is impossible to split the string any further.

Constraints:

1 <= s.length <= 16

s contains only lower case English letters.

==
Problem Number: 1204 URL: https://leetcode.com/problems/maximum-non-negative-product-in-a-matrix Title: 1594. Maximum Non Negative Product in a Matrix Problem Description: You are given a m x n matrix grid. Initially, you are located at the top-left corner (0, 0), and in each step, you can only move right or down in the matrix. Among all possible paths starting from the top-left corner (0, 0) and ending in the bottom-right corner (m - 1, n - 1), find the path with the maximum non-negative product. The product of a path is the product of all integers in the grid cells visited along the path. Return the maximum non-negative product modulo 109 + 7. If the maximum product is negative, return -1. Notice that the modulo is performed after getting the maximum product. Example 1:

Input: grid = [[-1,-2,-3],[-2,-3,-3],[-3,-3,-2]] Output: -1 Explanation: It is not possible to get non-negative product in the path from (0, 0) to (2, 2), so return -1.

Example 2:

Input: grid = [[1,-2,1],[1,-2,1],[3,-4,1]] Output: 8 Explanation: Maximum non-negative product is shown (1 * 1 * -2 * -4 * 1 = 8).

Example 3:

Input: grid = [[1,3],[0,-4]] Output: 0 Explanation: Maximum non-negative product is shown (1 * 0 * -4 = 0).

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 15 -4 <= grid[i][j] <= 4

==

Problem Number: 1205 URL: https://leetcode.com/problems/maximum-profit-of-operating-a-centennial-wheel Title: 1599. Maximum Profit of Operating a Centennial Wheel Problem Description: You are the operator of a Centennial Wheel that has four gondolas, and each gondola has room for up to four people. You have the ability to rotate the gondolas counterclockwise, which costs you runningCost dollars. You are given an array customers of length n where customers[i] is the number of new customers arriving just before the ith rotation (0-indexed). This means you must rotate the wheel i times before the customers[i] customers arrive. You cannot make customers wait if there is room in the gondola. Each customer pays boardingCost dollars when they board on the gondola closest to the ground and will exit once that gondola reaches the ground again. You can stop the wheel at any time, including before serving all customers. If you decide to stop serving customers, all subsequent rotations are free in order to get all the customers down safely. Note that if there are currently more than four customers waiting at the wheel, only four will board the gondola, and the rest will wait for the next rotation. Return the minimum number of rotations you need to perform to maximize your profit. If there is no scenario where the profit is positive, return -1.   Example 1:

Input: customers = [8,3], boardingCost = 5, runningCost = 6 Output: 3 Explanation: The numbers written on the gondolas are the number of people currently there. 1. 8 customers arrive, 4 board and 4 wait for the next gondola, the wheel rotates. Current profit is 4 * $5 - 1 * $6 = $14. 2. 3 customers arrive, the 4 waiting board the wheel and the other 3 wait, the wheel rotates. Current profit is 8 * $5 - 2 * $6 = $28. 3. The final 3 customers board the gondola, the wheel rotates. Current profit is 11 * $5 - 3 * $6 = $37. The highest profit was $37 after rotating the wheel 3 times.

Example 2: Input: customers = [10,9,6], boardingCost = 6, runningCost = 4 Output: 7 Explanation: 1. 10 customers arrive, 4 board and 6 wait for the next gondola, the wheel rotates. Current profit is 4 * $6 - 1 * $4 = $20. 2. 9 customers arrive, 4 board and 11 wait (2 originally waiting, 9 newly waiting), the wheel rotates. Current profit is 8 * $6 - 2 * $4 = $40. 3. The final 6 customers arrive, 4 board and 13 wait, the wheel rotates. Current profit is 12 * $6 - 3 * $4 = $60. 4. 4 board and 9 wait, the wheel rotates. Current profit is 16 * $6 - 4 * $4 = $80. 5. 4 board and 5 wait, the wheel rotates. Current profit is 20 * $6 - 5 * $4 = $100. 6. 4 board and 1 waits, the wheel rotates. Current profit is 24 * $6 - 6 * $4 = $120. 7. 1 boards, the wheel rotates. Current profit is 25 * $6 - 7 * $4 = $122. The highest profit was $122 after rotating the wheel 7 times.

Example 3: Input: customers = [3,4,0,5,1], boardingCost = 1, runningCost = 92 Output: -1 Explanation: 1. 3 customers arrive, 3 board and 0 wait, the wheel rotates. Current profit is 3 * $1 - 1 * $92 = -$89. 2. 4 customers arrive, 4 board and 0 wait, the wheel rotates. Current profit is 7 * $1 - 2 * $92 = -$177. 3. 0 customers arrive, 0 board and 0 wait, the wheel rotates. Current profit is 7 * $1 - 3 * $92 = -$269. 4. 5 customers arrive, 4 board and 1 waits, the wheel rotates. Current profit is 11 * $1 - 4 * $92 = -$357. 5. 1 customer arrives, 2 board and 0 wait, the wheel rotates. Current profit is 13 * $1 - 5 * $92 = -$447. The profit was never positive, so return -1.

Constraints:

n == customers.length 1 <= n <= 105 0 <= customers[i] <= 50 1 <= boardingCost, runningCost <= 100

==
Problem Number: 1206 URL: https://leetcode.com/problems/throne-inheritance Title: 1600. Throne Inheritance Problem Description: A kingdom consists of a king, his children, his grandchildren, and so on. Every once in a while, someone in the family dies or a child is born. The kingdom has a well-defined order of inheritance that consists of the king as the first member. Let's define the recursive function Successor(x, curOrder), which given a person x and the inheritance order so far, returns who should be the next person after x in the order of inheritance. Successor(x, curOrder): if x has no children or all of x's children are in curOrder: if x is the king return null else return Successor(x's parent, curOrder) else return x's oldest child who's not in curOrder

For example, assume we have a kingdom that consists of the king, his children Alice and Bob (Alice is older than Bob), and finally Alice's son Jack.

In the beginning, curOrder will be ["king"]. Calling Successor(king, curOrder) will return Alice, so we append to curOrder to get ["king", "Alice"]. Calling Successor(Alice, curOrder) will return Jack, so we append to curOrder to get ["king", "Alice", "Jack"]. Calling Successor(Jack, curOrder) will return Bob, so we append to curOrder to get ["king", "Alice", "Jack", "Bob"]. Calling Successor(Bob, curOrder) will return null. Thus the order of inheritance will be ["king", "Alice", "Jack", "Bob"].

Using the above function, we can always obtain a unique order of inheritance. Implement the ThroneInheritance class:

ThroneInheritance(string kingName) Initializes an object of the ThroneInheritance class. The name of the king is given as part of the constructor. void birth(string parentName, string childName) Indicates that parentName gave birth to childName. void death(string name) Indicates the death of name. The death of the person doesn't affect the Successor function nor the current inheritance order. You can treat it as just marking the person as dead. string[] getInheritanceOrder() Returns a list representing the current order of inheritance excluding dead people.

Example 1: Input ["ThroneInheritance", "birth", "birth", "birth", "birth",

"birth", "birth", "getInheritanceOrder", "death", "getInheritanceOrder"] [["king"], ["king", "andy"], ["king", "bob"], ["king", "catherine"], ["andy", "matthew"], ["bob", "alex"], ["bob", "asha"], [null], ["bob"], [null]] Output [null, null, null, null, null, null, null, ["king", "andy", "matthew", "bob", "alex", "asha", "catherine"], null, ["king", "andy", "matthew", "alex", "asha", "catherine"]]

Explanation ThroneInheritance t= new ThroneInheritance("king"); // order: king t.birth("king", "andy"); // order: king > andy t.birth("king", "bob"); // order: king > andy > bob t.birth("king", "catherine"); // order: king > andy > bob > catherine t.birth("andy", "matthew"); // order: king > andy > matthew > bob > catherine t.birth("bob", "alex"); // order: king > andy > matthew > bob > alex > catherine t.birth("bob", "asha"); // order: king > andy > matthew > bob > alex > asha > catherine t.getInheritanceOrder(); // return ["king", "andy", "matthew", "bob", "alex", "asha", "catherine"] t.death("bob"); // order: king > andy > matthew > bob > alex > asha > catherine t.getInheritanceOrder(); // return ["king", "andy", "matthew", "alex", "asha", "catherine"]

Constraints:

1 <= kingName.length, parentName.length, childName.length, name.length <= 15 kingName, parentName, childName, and name consist of lowercase English letters only. All arguments childName and kingName are distinct. All name arguments of death will be passed to either the constructor or as childName to birth first. For each call to birth(parentName, childName), it is guaranteed that parentName is alive. At most 105 calls will be made to birth and death. At most 10 calls will be made to getInheritanceOrder.

==

Problem Number: 1207 URL: https://leetcode.com/problems/alert-using-same-key-card-three-or-more-times-in-a-one-hour-period Title: 1604. Alert Using Same Key-Card Three or More Times in a One Hour Period Problem Description: LeetCode company workers use key-cards to unlock office doors. Each time a worker uses their key-card, the security system saves the worker's name and the time when it was used. The system emits an alert if any worker uses the key-card three or more times in a one-hour period. You are given a list of strings keyName and keyTime where [keyName[i], keyTime[i]] corresponds to a person's name and the time when their key-card was used in a single day. Access times are given in the 24-hour time format "HH:MM", such as "23:51" and "09:49". Return a list of unique worker names who received an alert for frequent keycard use. Sort the names in ascending order alphabetically. Notice that "10:00" - "11:00" is considered to be within a one-hour period, while "22:51" - "23:52" is not considered to be within a one-hour period. Example 1: Input: keyName = ["daniel","daniel","daniel","luis","luis","luis","luis"], keyTime = ["10:00","10:40","11:00","09:00","11:00","13:00","15:00"] Output: ["daniel"] Explanation: "daniel" used the keycard 3 times in a one-hour period ("10:00","10:40", "11:00").

Example 2: Input: keyName = ["alice","alice","alice","bob","bob","bob","bob"], keyTime = ["12:01","12:00","18:00","21:00","21:20","21:30","23:00"] Output: ["bob"] Explanation: "bob" used the keycard 3 times in a one-hour period ("21:00","21:20", "21:30").

Constraints:

1 <= keyName.length, keyTime.length <= 105 keyName.length == keyTime.length keyTime[i] is in the format "HH:MM". [keyName[i], keyTime[i]] is unique. 1 <= keyName[i].length <= 10 keyName[i] contains only lowercase English letters.

==

Problem Number: 1208 URL: https://leetcode.com/problems/find-valid-matrix-given-row-and-column-sums Title: 1605. Find Valid Matrix Given Row and Column Sums Problem Description: You are given two arrays rowSum and colSum of non-negative integers where rowSum[i] is the sum of the elements in the ith row and colSum[j] is the sum of the elements of the jth column of a 2D matrix. In other words, you do not know the elements of the matrix, but you do know the sums of each row and column. Find any matrix of non-negative integers of size rowSum.length x colSum.length that satisfies the rowSum and colSum requirements. Return a 2D array representing any matrix that fulfills the requirements. It's guaranteed that at least one matrix that fulfills the requirements exists. Example 1: Input: rowSum = [3,8], colSum = [4,7] Output: [[3,0], [1,7]] Explanation: 0th row: 3 + 0 = 3 == rowSum[0] 1st row: 1 + 7 = 8 == rowSum[1] 0th column: 3 + 1 = 4 == colSum[0] 1st column: 0 + 7 = 7 == colSum[1] The row and column sums match, and all matrix elements are non-negative. Another possible matrix is: [[1,2], [3,5]]

Example 2: Input: rowSum = [5,7,10], colSum = [8,6,8] Output: [[0,5,0], [6,1,0], [2,0,8]]

Constraints:

1 <= rowSum.length, colSum.length <= 500 0 <= rowSum[i], colSum[i] <= 108 sum(rowSum) == sum(colSum)

==

Problem Number: 1209 URL: https://leetcode.com/problems/even-odd-tree Title: 1609. Even Odd Tree Problem Description: A binary tree is named Even-Odd if it meets the following conditions:

The root of the binary tree is at level index 0, its children are at level index 1, their children are at level index 2, etc. For every even-indexed level, all nodes at the level have odd integer values in strictly increasing order (from left to right). For every odd-indexed level, all nodes at the level have even integer values in strictly decreasing order (from left to right).

Given the root of a binary tree, return true if the binary tree is Even-Odd, otherwise return false. Example 1:

Input: root = [1,10,4,3,null,7,9,12,8,6,null,null,2] Output: true Explanation: The node values on each level are: Level 0: [1] Level 1: [10,4] Level 2: [3,7,9] Level 3: [12,8,6,2] Since levels 0 and 2 are all odd and increasing and levels 1 and 3 are all even and decreasing, the tree is Even-Odd.

Example 2:

Input: root = [5,4,2,3,3,7] Output: false Explanation: The node values on each level are: Level 0: [5] Level 1: [4,2] Level 2: [3,3,7] Node values in level 2 must be in strictly increasing order, so the tree is not Even-Odd.

Example 3:

Input: root = [5,9,1,3,5,7] Output: false Explanation: Node values in the level 1 should be even integers.

Constraints:

The number of nodes in the tree is in the range [1, 105]. 1 <= Node.val <= 106

==

Problem Number: 1210 URL: https://leetcode.com/problems/maximal-network-rank Title: 1615. Maximal Network Rank Problem Description: There is an infrastructure of n cities with some number of roads connecting these cities. Each roads[i] = [ai, bi] indicates that there is a bidirectional road between cities ai and bi. The network rank of two different cities is defined as the total number of directly connected roads to either city. If a road is directly connected to both cities, it is only counted once. The maximal network rank of the infrastructure is the maximum network rank of all pairs of different cities. Given the integer n and the array roads, return the maximal network rank of the entire infrastructure. Example 1:

Input: n = 4, roads = [[0,1],[0,3],[1,2],[1,3]] Output: 4 Explanation: The network rank of cities 0 and 1 is 4 as there are 4 roads that are connected to either 0 or 1. The road between 0 and 1 is only counted once.

Example 2:

Input: n = 5, roads = [[0,1],[0,3],[1,2],[1,3],[2,3],[2,4]] Output: 5 Explanation: There are 5 roads that are connected to cities 1 or 2.

Example 3: Input: n = 8, roads = [[0,1],[1,2],[2,3],[2,4],[5,6],[5,7]] Output: 5 Explanation: The network rank of 2 and 5 is 5. Notice that all the cities do not have to be connected.

Constraints:

2 <= n <= 100 0 <= roads.length <= n * (n - 1) / 2 roads[i].length == 2 0 <= ai, bi <= n-1 ai != bi Each pair of cities has at most one road connecting them.

==
Problem Number: 1211 URL: https://leetcode.com/problems/split-two-strings-to-make-palindrome Title: 1616. Split Two Strings to Make Palindrome Problem Description: You are given two strings a and b of the same length. Choose an index and split both strings at the same index, splitting a into two strings: aprefix and asuffix where a = aprefix + asuffix, and splitting b into two strings: bprefix and bsuffix where b = bprefix + bsuffix. Check if aprefix + bsuffix or bprefix + asuffix forms a palindrome. When you split a string s into sprefix and ssuffix, either ssuffix or sprefix is allowed to be empty. For example, if s = "abc", then "" + "abc", "a" + "bc", "ab" + "c" , and "abc" + "" are valid splits. Return true if it is possible to form a palindrome string, otherwise return false. Notice that x + y denotes the concatenation of strings x and y. Example 1: Input: a = "x", b = "y" Output: true Explaination: If either a or b are palindromes the answer is true since you can split in the following way: aprefix = "", asuffix = "x" bprefix = "", bsuffix = "y" Then, aprefix + bsuffix = "" + "y" = "y", which is a palindrome.

Example 2: Input: a = "xbdef", b = "xecab" Output: false

Example 3: Input: a = "ulacfd", b = "jizalu" Output: true Explaination: Split them at index 3: aprefix = "ula", asuffix = "cfd" bprefix = "jiz", bsuffix = "alu" Then, aprefix + bsuffix = "ula" + "alu" = "ulaalu", which is a palindrome.

Constraints:

1 <= a.length, b.length <= 105 a.length == b.length a and b consist of lowercase English letters

==
Problem Number: 1212 URL: https://leetcode.com/problems/coordinate-with-maximum-network-quality Title: 1620. Coordinate With Maximum Network Quality Problem Description: You are given an array of network towers towers, where towers[i] = [xi, yi, qi] denotes the ith network tower with location (xi, yi) and quality factor qi. All the coordinates are integral coordinates on the X-Y plane, and the distance between the two coordinates is the Euclidean distance. You are also given an integer radius where a tower is reachable if the distance is less than or equal to radius. Outside that distance, the signal becomes garbled, and the tower is not reachable. The signal quality of the ith tower at a coordinate (x, y) is calculated with the formula qi / (1 + d) , where d is the distance between the tower and the coordinate. The network quality at a coordinate is the sum of the signal qualities from all the reachable towers. Return the array [cx, cy] representing the integral coordinate (cx, cy) where the network quality is maximum. If there are multiple coordinates with the same network quality, return the lexicographically minimum non-negative coordinate. Note:

A coordinate (x1, y1) is lexicographically smaller than (x2, y2) if either:

x1 < x2, or x1 == x2 and y1 < y2.

val is the greatest integer less than or equal to val (the floor function).

Example 1:

Input: towers = [[1,2,5],[2,1,7],[3,1,9]], radius = 2 Output: [2,1] Explanation: At coordinate (2, 1) the total quality is 13. - Quality of 7 from (2, 1) results in 7 / (1 + sqrt(0) = 7 = 7 - Quality of 5 from (1, 2) results in 5 / (1 + sqrt(2) = 2.07 = 2 - Quality of 9 from (3, 1) results in 9 / (1 + sqrt(1) = 4.5 = 4 No other coordinate has a higher network quality. Example 2: Input: towers = [[23,11,21]], radius = 9 Output: [23,11] Explanation: Since there is only one tower, the network quality is highest right at the tower's location.

Example 3: Input: towers = [[1,2,13],[2,1,7],[0,1,9]], radius = 2 Output: [1,2] Explanation: Coordinate (1, 2) has the highest network quality.

Constraints:

1 <= towers.length <= 50 towers[i].length == 3 0 <= xi, yi, qi <= 50 1 <= radius <= 50

==

Problem Number: 1213 URL: https://leetcode.com/problems/number-of-sets-of-k-non-overlapping-line-segments Title: 1621. Number of Sets of K Non-Overlapping Line Segments Problem Description: Given n points on a 1-D plane, where the ith point (from 0 to n-1) is at x = i, find the number of ways we can draw exactly k non-overlapping line segments such that each segment covers two or more points. The endpoints of each segment must have integral coordinates. The k line segments do not have to cover all n points, and they are allowed to share endpoints. Return the number of ways we can draw k non-overlapping line segments. Since this number can be huge, return it modulo 109 + 7. Example 1:

Input: n = 4, k = 2 Output: 5 Explanation: The two line segments are shown in red and blue. The image above shows the 5 different ways {(0,2),(2,3)}, {(0,1),(1,3)}, {(0,1),(2,3)}, {(1,2),(2,3)}, {(0,1),(1,2)}.

Example 2: Input: n = 3, k = 1 Output: 3 Explanation: The 3 ways are {(0,1)}, {(0,2)}, {(1,2)}.

Example 3: Input: n = 30, k = 7 Output: 796297179 Explanation: The total number of possible ways to draw 7 line segments is 3796297200. Taking this number modulo 109 + 7 gives us 796297179.

Constraints:

2 <= n <= 1000 1 <= k <= n-1

==

Problem Number: 1214 URL: https://leetcode.com/problems/lexicographically-smallest-string-after-applying-operations Title: 1625. Lexicographically Smallest String After Applying Operations Problem Description: You are given a string s of even length consisting of digits from 0 to 9, and two integers a and

b. You can apply either of the following two operations any number of times and in any order on s:

Add a to all odd indices of s (0-indexed). Digits post 9 are cycled back to 0. For example, if s = "3456" and a = 5, s becomes "3951". Rotate s to the right by b positions. For example, if s = "3456" and b = 1, s becomes "6345".

Return the lexicographically smallest string you can obtain by applying the above operations any number of times on s. A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b. For example, "0158" is lexicographically smaller than "0190" because the first position they differ is at the third letter, and '5' comes before '9'. Example 1: Input: s = "5525", a = 9, b = 2 Output: "2050" Explanation: We can apply the following operations: Start: "5525" Rotate: "2555" Add: "2454" Add: "2353" Rotate: "5323" Add: "5222" Add: "5121" Rotate: "2151" Add: "2050" There is no way to obtain a string that is lexicographically smaller than "2050".

Example 2: Input: s = "74", a = 5, b = 1 Output: "24" Explanation: We can apply the following operations: Start: "74" Rotate: "47" Add: "42" Rotate: "24" There is no way to obtain a string that is lexicographically smaller than "24".

Example 3: Input: s = "0011", a = 4, b = 2 Output: "0011" Explanation: There are no sequence of operations that will give us a lexicographically smaller string than "0011".

Constraints:

2 <= s.length <= 100 s.length is even. s consists of digits from 0 to 9 only. 1 <= a <= 9 1 <= b <= s.length - 1

==

Problem Number: 1215 URL: https://leetcode.com/problems/best-team-with-no-conflicts Title: 1626. Best Team With No Conflicts Problem Description: You are the manager of a basketball team. For the upcoming tournament, you want to choose the team with the highest overall score. The score of the team is the sum of scores of all the players in the team. However, the basketball team is not allowed to have conflicts. A conflict exists if a younger player has a strictly higher score than an older player. A conflict does not occur between players of the same age. Given two lists, scores and ages, where each scores[i] and ages[i] represents the score and age of the ith player, respectively, return the highest overall score of all possible basketball teams. Example 1: Input: scores = [1,3,5,10,15], ages = [1,2,3,4,5] Output: 34 Explanation: You can choose all the players.

Example 2: Input: scores = [4,5,6,5], ages = [2,1,2,1] Output: 16 Explanation: It is best to choose the last 3 players. Notice that you are allowed to choose multiple people of the same age.

Example 3: Input: scores = [1,2,3,5], ages = [8,9,10,1] Output: 6 Explanation: It is best to choose the first 3 players.

Constraints:

1 <= scores.length, ages.length <= 1000 scores.length == ages.length 1 <= scores[i] <= 106 1 <= ages[i] <= 1000

==

Problem Number: 1216 URL: https://leetcode.com/problems/arithmetic-subarrays Title: 1630. Arithmetic Subarrays Problem Description: A sequence of numbers is called arithmetic if it consists of at least two elements, and the difference between every two consecutive elements is the same. More formally, a sequence s is arithmetic if and only if s[i+1] - s[i] == s[1] - s[0] for all valid i. For example, these are arithmetic sequences: 1, 3, 5, 7, 9 7, 7, 7, 7 3, -1, -5, -9 The following sequence is not arithmetic: 1, 1, 2, 5, 7 You are given an array of n integers, nums, and two arrays of m integers each, l and r, representing the m range queries, where the ith query is the range [l[i], r[i]]. All the arrays are 0-indexed. Return a list of boolean elements answer, where answer[i] is true if the subarray nums[l[i]], nums[l[i]+1], ... , nums[r[i]] can be rearranged to form an arithmetic sequence, and false otherwise. Example 1: Input: nums = [4,6,5,9,3,7], l = [0,0,2], r = [2,3,5] Output: [true,false,true] Explanation: In the 0th query, the subarray is [4,6,5]. This can be rearranged as [6,5,4], which is an arithmetic sequence. In the 1st query, the subarray is [4,6,5,9]. This cannot be rearranged as an arithmetic sequence. In the 2nd query, the subarray is [5,9,3,7]. This can be rearranged as [3,5,7,9], which is an arithmetic sequence. Example 2: Input: nums = [-12,-9,-3,-12,-6,15,20,-25,-20,-15,-10], l = [0,1,6,4,8,7], r = [4,4,9,7,9,10] Output: [false,true,false,false,true,true]

Constraints:

n == nums.length m == l.length m == r.length 2 <= n <= 500 1 <= m <= 500 0 <= l[i] < r[i] < n -105 <= nums[i] <= 105

==

Problem Number: 1217 URL: https://leetcode.com/problems/path-with-minimum-effort Title: 1631. Path With Minimum Effort Problem Description: You are a hiker preparing for an upcoming hike. You are given heights, a 2D array of size rows x columns, where heights[row][col] represents the height of cell (row, col). You are situated in the top-left cell, (0, 0), and you hope to travel to the bottom-right cell, (rows-1, columns-1) (i.e., 0-indexed). You can move up, down, left, or right, and you wish to find a route that requires the minimum effort. A route's effort is the maximum absolute difference in heights between two consecutive cells of the route. Return the minimum effort required to travel from the top-left cell to the bottom-right cell. Example 1:

Input: heights = [[1,2,2],[3,8,2],[5,3,5]] Output: 2 Explanation: The route of [1,3,5,3,5] has a maximum absolute difference of 2 in consecutive cells. This is better than the route of [1,2,2,2,5], where the maximum absolute difference is

3.

Example 2:

Input: heights = [[1,2,3],[3,8,4],[5,3,5]] Output: 1 Explanation: The route of [1,2,3,4,5] has a maximum absolute difference of 1 in consecutive cells, which is better than route [1,3,5,3,5].

Example 3:

Input: heights = [[1,2,1,1,1],[1,2,1,2,1],[1,2,1,2,1],[1,2,1,2,1],[1,1,1,2,1]] Output: 0 Explanation: This route does not require any effort.

Constraints:

rows == heights.length columns == heights[i].length 1 <= rows, columns <= 100 1 <= heights[i][j] <= 106

==
Problem Number: 1218 URL: https://leetcode.com/problems/widest-vertical-area-between-two-points-containing-no-points Title: 1637. Widest Vertical Area Between Two Points Containing No Points Problem Description: Given n points on a 2D plane where points[i] = [xi, yi], Return the widest vertical area between two points such that no points are inside the area. A vertical area is an area of fixed-width extending infinitely along the y-axis (i.e., infinite height). The widest vertical area is the one with the maximum width. Note that points on the edge of a vertical area are not considered included in the area. Example 1: Input: points = [[8,7],[9,9],[7,4],[9,7]] Output: 1 Explanation: Both the red and the blue area are optimal.

Example 2: Input: points = [[3,1],[9,0],[1,0],[1,4],[5,3],[8,8]] Output: 3

Constraints:

n == points.length 2 <= n <= 105 points[i].length == 2 0 <= xi, yi <= 109

==
Problem Number: 1219 URL: https://leetcode.com/problems/count-substrings-that-differ-by-one-character Title: 1638. Count Substrings That Differ by One Character Problem Description: Given two strings s and t, find the number of ways you can choose a non-empty substring of s and replace a single character by a different character such that the resulting substring is a substring of t. In other words, find the number of substrings in s that differ from some substring in t by exactly one character. For example, the underlined substrings in "computer" and "computation" only differ by the 'e'/'a', so this is a valid way. Return the number of substrings that satisfy the condition above. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "aba", t = "baba" Output: 6 Explanation: The following are the pairs of substrings from s and t that differ by exactly 1 character: ("aba", "baba") ("aba", "baba") ("aba", "baba") ("aba", "baba") ("aba", "baba")

("aba", "baba") The underlined portions are the substrings that are chosen from s and t.

Example 2: Input: s = "ab", t = "bb" Output: 3 Explanation: The following are the pairs of substrings from s and t that differ by 1 character: ("ab", "bb") ("ab", "bb") ("ab", "bb") The underlined portions are the substrings that are chosen from s and t.

Constraints:

1 <= s.length, t.length <= 100 s and t consist of lowercase English letters only.

==
Problem Number: 1220 URL: https://leetcode.com/problems/count-sorted-vowel-strings Title: 1641. Count Sorted Vowel Strings Problem Description: Given an integer n, return the number of strings of length n that consist only of vowels (a, e, i, o, u) and are lexicographically sorted. A string s is lexicographically sorted if for all valid i, s[i] is the same as or comes before s[i+1] in the alphabet. Example 1: Input: n = 1 Output: 5 Explanation: The 5 sorted strings that consist of vowels only are ["a","e","i","o","u"].

Example 2: Input: n = 2 Output: 15 Explanation: The 15 sorted strings that consist of vowels only are ["aa","ae","ai","ao","au","ee","ei","eo","eu","ii","io","iu","oo","ou","uu"]. Note that "ea" is not a valid string since 'e' comes after 'a' in the alphabet.

Example 3: Input: n = 33 Output: 66045

Constraints:

1 <= n <= 50

==
Problem Number: 1221 URL: https://leetcode.com/problems/furthest-building-you-can-reach Title: 1642. Furthest Building You Can Reach Problem Description: You are given an integer array heights representing the heights of buildings, some bricks, and some ladders. You start your journey from building 0 and move to the next building by possibly using bricks or ladders. While moving from building i to building i+1 (0-indexed),

If the current building's height is greater than or equal to the next building's height, you do not need a ladder or bricks. If the current building's height is less than the next building's height, you can either use one ladder or (h[i+1] - h[i]) bricks.

Return the furthest building index (0-indexed) you can reach if you use the given ladders and bricks optimally. Example 1:

Input: heights = [4,2,7,6,9,14,12], bricks = 5, ladders = 1 Output: 4 Explanation: Starting at building 0, you can follow these steps: - Go to building 1 without using ladders nor bricks since 4 >= 2. - Go to building 2 using 5 bricks. You must use either bricks or ladders because 2 < 7. - Go to building 3 without using ladders nor bricks since 7 >= 6. - Go to building 4 using your only ladder.

You must use either bricks or ladders because 6 < 9. It is impossible to go beyond building 4 because you do not have any more bricks or ladders.

Example 2: Input: heights = [4,12,2,7,3,18,20,3,19], bricks = 10, ladders = 2 Output: 7

Example 3: Input: heights = [14,3,19,3], bricks = 17, ladders = 0 Output: 3

Constraints:

1 <= heights.length <= 105 1 <= heights[i] <= 106 0 <= bricks <= 109 0 <= ladders <= heights.length

==
Problem Number: 1222 URL: https://leetcode.com/problems/minimum-deletions-to-make-character-frequencies-unique Title: 1647. Minimum Deletions to Make Character Frequencies Unique Problem Description: A string s is called good if there are no two different characters in s that have the same frequency. Given a string s, return the minimum number of characters you need to delete to make s good. The frequency of a character in a string is the number of times it appears in the string. For example, in the string "aab", the frequency of 'a' is 2, while the frequency of 'b' is 1. Example 1: Input: s = "aab" Output: 0 Explanation: s is already good.

Example 2: Input: s = "aaabbbcc" Output: 2 Explanation: You can delete two 'b's resulting in the good string "aaabcc". Another way it to delete one 'b' and one 'c' resulting in the good string "aaabbc". Example 3: Input: s = "ceabaacb" Output: 2 Explanation: You can delete both 'c's resulting in the good string "eabaab". Note that we only care about characters that are still in the string at the end (i.e. frequency of 0 is ignored).

Constraints:

1 <= s.length <= 105 s contains only lowercase English letters.

==
Problem Number: 1223 URL: https://leetcode.com/problems/sell-diminishing-valued-colored-balls Title: 1648. Sell Diminishing-Valued Colored Balls Problem Description: You have an inventory of different colored balls, and there is a customer that wants orders balls of any color. The customer weirdly values the colored balls. Each colored ball's value is the number of balls of that color you currently have in your inventory. For example, if you own 6 yellow balls, the customer would pay 6 for the first yellow ball. After the transaction, there are only 5 yellow balls left, so the next yellow ball is then valued at 5 (i.e., the value of the balls decreases as you sell more to the customer). You are given an integer array, inventory, where inventory[i] represents the number of balls of the ith color that you initially own. You are also given an integer orders, which represents the total number of balls that the customer wants. You can sell the balls in any order. Return the maximum total value that you

can attain after selling orders colored balls. As the answer may be too large, return it modulo 109 + 7. Example 1:

Input: inventory = [2,5], orders = 4 Output: 14 Explanation: Sell the 1st color 1 time (2) and the 2nd color 3 times (5 + 4 + 3). The maximum total value is 2 + 5 + 4 + 3 = 14.

Example 2: Input: inventory = [3,5], orders = 6 Output: 19 Explanation: Sell the 1st color 2 times (3 + 2) and the 2nd color 4 times (5 + 4 + 3 + 2). The maximum total value is 3 + 2 + 5 + 4 + 3 + 2 = 19.

Constraints:

1 <= inventory.length <= 105 1 <= inventory[i] <= 109 1 <= orders <= min(sum(inventory[i]), 109)

==
Problem Number: 1224 URL: https://leetcode.com/problems/minimum-deletions-to-make-string-balanced Title: 1653. Minimum Deletions to Make String Balanced Problem Description: You are given a string s consisting only of characters 'a' and 'b'. You can delete any number of characters in s to make s balanced. s is balanced if there is no pair of indices (i,j) such that i < j and s[i] = 'b' and s[j]= 'a'. Return the minimum number of deletions needed to make s balanced. Example 1: Input: s = "aababbab" Output: 2 Explanation: You can either: Delete the characters at 0-indexed positions 2 and 6 ("aababbab" -> "aaabbb"), or Delete the characters at 0-indexed positions 3 and 6 ("aababbab" -> "aabbbb").

Example 2: Input: s = "bbaaaaabb" Output: 2 Explanation: The only solution is to delete the first two characters.

Constraints:

1 <= s.length <= 105 s[i] is 'a' or 'b'.

==
Problem Number: 1225 URL: https://leetcode.com/problems/minimum-jumps-to-reach-home Title: 1654. Minimum Jumps to Reach Home Problem Description: A certain bug's home is on the x-axis at position x. Help them get there from position 0. The bug jumps according to the following rules:

It can jump exactly a positions forward (to the right). It can jump exactly b positions backward (to the left). It cannot jump backward twice in a row. It cannot jump to any forbidden positions.

The bug may jump forward beyond its home, but it cannot jump to positions numbered with negative integers. Given an array of integers forbidden, where forbidden[i] means that the bug cannot jump to the position forbidden[i], and integers a, b, and x, return the minimum number of jumps needed for the bug to reach its home. If there is no possible sequence of jumps that lands the bug on position x, return -1. Example 1: Input: forbidden = [14,4,18,1,15], a = 3, b = 15, x = 9 Output: 3 Explanation: 3 jumps forward (0 -> 3 -> 6 -> 9) will get the bug home.

Example 2: Input: forbidden = [8,3,16,6,12,20], a = 15, b = 13, x = 11 Output: -1

Example 3: Input: forbidden = [1,6,2,14,5,17,4], a = 16, b = 9, x = 7 Output: 2 Explanation: One jump forward (0 -> 16) then one jump backward (16 -> 7) will get the bug home.

Constraints:

1 <= forbidden.length <= 1000 1 <= a, b, forbidden[i] <= 2000 0 <= x <= 2000 All the elements in forbidden are distinct. Position x is not forbidden.

==
Problem Number: 1226 URL: https://leetcode.com/problems/determine-if-two-strings-are-close Title: 1657. Determine if Two Strings Are Close Problem Description: Two strings are considered close if you can attain one from the other using the following operations:

Operation 1: Swap any two existing characters.

For example, abcde -> aecdb

Operation 2: Transform every occurrence of one existing character into another existing character, and do the same with the other character.

For example, aacabb -> bbcbaa (all a's turn into b's, and all b's turn into a's)

You can use the operations on either string as many times as necessary. Given two strings, word1 and word2, return true if word1 and word2 are close, and false otherwise. Example 1: Input: word1 = "abc", word2 = "bca" Output: true Explanation: You can attain word2 from word1 in 2 operations. Apply Operation 1: "abc" -> "acb" Apply Operation 1: "acb" -> "bca"

Example 2: Input: word1 = "a", word2 = "aa" Output: false Explanation: It is impossible to attain word2 from word1, or vice versa, in any number of operations.

Example 3: Input: word1 = "cabbba", word2 = "abbccc" Output: true Explanation: You can attain word2 from word1 in 3 operations. Apply Operation 1: "cabbba" -> "caabbb" Apply Operation 2: "caabbb" -> "baaccc" Apply Operation 2: "baaccc" -> "abbccc"

Constraints:

1 <= word1.length, word2.length <= 105 word1 and word2 contain only lowercase English letters.

==
Problem Number: 1227 URL: https://leetcode.com/problems/minimum-operations-to-reduce-x-to-zero Title: 1658. Minimum Operations to Reduce

X to Zero Problem Description: You are given an integer array nums and an integer x. In one operation, you can either remove the leftmost or the rightmost element from the array nums and subtract its value from x. Note that this modifies the array for future operations. Return the minimum number of operations to reduce x to exactly 0 if it is possible, otherwise, return -1. Example 1: Input: nums = [1,1,4,2,3], x = 5 Output: 2 Explanation: The optimal solution is to remove the last two elements to reduce x to zero.

Example 2: Input: nums = [5,6,7,8,9], x = 4 Output: -1

Example 3: Input: nums = [3,2,20,1,1,3], x = 10 Output: 5 Explanation: The optimal solution is to remove the last three elements and the first two elements (5 operations in total) to reduce x to zero.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 104 1 <= x <= 109

==

Problem Number: 1228 URL: https://leetcode.com/problems/smallest-string-with-a-given-numeric-value Title: 1663. Smallest String With A Given Numeric Value Problem Description: The numeric value of a lowercase character is defined as its position (1-indexed) in the alphabet, so the numeric value of a is 1, the numeric value of b is 2, the numeric value of c is 3, and so on. The numeric value of a string consisting of lowercase characters is defined as the sum of its characters' numeric values. For example, the numeric value of the string "abe" is equal to 1 + 2 + 5 = 8. You are given two integers n and k. Return the lexicographically smallest string with length equal to n and numeric value equal to k. Note that a string x is lexicographically smaller than string y if x comes before y in dictionary order, that is, either x is a prefix of y, or if i is the first position such that x[i] != y[i], then x[i] comes before y[i] in alphabetic order. Example 1: Input: n = 3, k = 27 Output: "aay" Explanation: The numeric value of the string is 1 + 1 + 25 = 27, and it is the smallest string with such a value and length equal to 3.

Example 2: Input: n = 5, k = 73 Output: "aaszz"

Constraints:

1 <= n <= 105 n <= k <= 26 * n

==

Problem Number: 1229 URL: https://leetcode.com/problems/ways-to-make-a-fair-array Title: 1664. Ways to Make a Fair Array Problem Description: You are given an integer array nums. You can choose exactly one index (0-indexed) and remove the element. Notice that the index of the elements may change after the removal. For example, if nums = [6,1,7,4,1]:

Choosing to remove index 1 results in nums = [6,7,4,1]. Choosing to remove index 2 results in nums = [6,1,4,1]. Choosing to remove index 4 results in nums = [6,1,7,4].

An array is fair if the sum of the odd-indexed values equals the sum of the even-indexed values. Return the number of indices that you could choose such that after the removal, nums is fair.   Example 1: Input: nums = [2,1,6,4] Output: 1 Explanation: Remove index 0: [1,6,4] -> Even sum: 1 + 4 = 5. Odd sum: 6. Not fair. Remove index 1: [2,6,4] -> Even sum: 2 + 4 = 6. Odd sum: 6. Fair. Remove index 2: [2,1,4] -> Even sum: 2 + 4 = 6. Odd sum: 1. Not fair. Remove index 3: [2,1,6] -> Even sum: 2 + 6 = 8. Odd sum: 1. Not fair. There is 1 index that you can remove to make nums fair.

Example 2: Input: nums = [1,1,1] Output: 3 Explanation: You can remove any index and the remaining array is fair.

Example 3: Input: nums = [1,2,3] Output: 0 Explanation: You cannot make a fair array after removing any index.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 104

==
Problem Number: 1230 URL: https://leetcode.com/problems/merge-in-between-linked-lists Title: 1669. Merge In Between Linked Lists Problem Description: You are given two linked lists: list1 and list2 of sizes n and m respectively. Remove list1's nodes from the ath node to the bth node, and put list2 in their place. The blue edges and nodes in the following figure indicate the result:

Build the result list and return its head.   Example 1:

Input: list1 = [0,1,2,3,4,5], a = 3, b = 4, list2 = [1000000,1000001,1000002] Output: [0,1,2,1000000,1000001,1000002,5] Explanation: We remove the nodes 3 and 4 and put the entire list2 in their place. The blue edges and nodes in the above figure indicate the result.

Example 2:

Input: list1 = [0,1,2,3,4,5,6], a = 2, b = 5, list2 = [1000000,1000001,1000002,1000003,1000004] Output: [0,1,1000000,1000001,1000002,1000003,1000004,6] Explanation: The blue edges and nodes in the above figure indicate the result.

Constraints:

3 <= list1.length <= 104 1 <= a <= b < list1.length - 1 1 <= list2.length <= 104

==
Problem Number: 1231 URL: https://leetcode.com/problems/design-front-middle-back-queue Title: 1670. Design Front Middle Back Queue Problem Description: Design a queue that supports push and pop operations in the front, middle, and back. Implement the FrontMiddleBack class:

FrontMiddleBack() Initializes the queue. void pushFront(int val) Adds val to the front of the queue. void pushMiddle(int val) Adds val to the middle of the queue. void pushBack(int val) Adds val to the back of the queue. int popFront() Removes the front element of the queue and returns it. If the queue is empty, return -1. int popMiddle() Removes the middle element of the queue and returns it. If the queue is empty, return -1. int popBack() Removes the back element of the queue and returns it. If the queue is empty, return -1.

Notice that when there are two middle position choices, the operation is performed on the frontmost middle position choice. For example:

Pushing 6 into the middle of [1, 2, 3, 4, 5] results in [1, 2, 6, 3, 4, 5]. Popping the middle from [1, 2, 3, 4, 5, 6] returns 3 and results in [1, 2, 4, 5, 6].

Example 1: Input: ["FrontMiddleBackQueue", "pushFront", "pushBack", "pushMiddle", "pushMiddle", "popFront", "popMiddle", "popMiddle", "popBack", "popFront"] [[], [1], [2], [3], [4], [], [], [], [], []] Output: [null, null, null, null, null, 1, 3, 4, 2, -1]

Explanation: FrontMiddleBackQueue q = new FrontMiddleBackQueue(); q.pushFront(1); // [1] q.pushBack(2); // [1, 2] q.pushMiddle(3); // [1, 3, 2] q.pushMiddle(4); // [1, 4, 3, 2] q.popFront(); // return 1 -> [4, 3, 2] q.popMiddle(); // return 3 -> [4, 2] q.popMiddle(); // return 4 -> [2] q.popBack(); // return 2 -> [] q.popFront(); // return -1 -> [] (The queue is empty)

Constraints:

1 <= val <= 109 At most 1000 calls will be made to pushFront, pushMiddle, pushBack, popFront, popMiddle, and popBack.

==
Problem Number: 1232 URL: https://leetcode.com/problems/find-the-most-competitive-subsequence Title: 1673. Find the Most Competitive Subsequence Problem Description: Given an integer array nums and a positive integer k, return the most competitive subsequence of nums of size k. An array's subsequence is a resulting sequence obtained by erasing some (possibly zero) elements from the array. We define that a subsequence a is more competitive than a subsequence b (of the same length) if in the first position where a and b differ, subsequence a has a number less than the corresponding number in b. For example, [1,3,4] is more competitive than [1,3,5] because the first position they differ is at the final number, and 4 is less than 5. Example 1: Input: nums = [3,5,2,6], k = 2 Output: [2,6] Explanation: Among the set of every possible subsequence: {[3,5], [3,2], [3,6], [5,2], [5,6], [2,6]}, [2,6] is the most competitive.

Example 2: Input: nums = [2,4,3,3,5,4,9,6], k = 4 Output: [2,3,3,4]

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109 1 <= k <= nums.length

== Problem Number: 1233 URL: https://leetcode.com/problems/minimum-moves-to-make-array-complementary Title: 1674. Minimum Moves to Make Array Complementary Problem Description: You are given an integer array nums of even length n and an integer limit. In one move, you can replace any integer from nums with another integer between 1 and limit, inclusive. The array nums is complementary if for all indices i (0-indexed), nums[i] + nums[n - 1 - i] equals the same number. For example, the array [1,2,3,4] is complementary because for all indices i, nums[i] + nums[n - 1 - i] = 5. Return the minimum number of moves required to make nums complementary. Example 1: Input: nums = [1,2,4,3], limit = 4 Output: 1 Explanation: In 1 move, you can change nums to [1,2,2,3] (underlined elements are changed). nums[0] + nums[3] = 1 + 3 = 4. nums[1] + nums[2] = 2 + 2 = 4. nums[2] + nums[1] = 2 + 2 = 4. nums[3] + nums[0] = 3 + 1 = 4. Therefore, nums[i] + nums[n-1-i] = 4 for every i, so nums is complementary.

Example 2: Input: nums = [1,2,2,1], limit = 2 Output: 2 Explanation: In 2 moves, you can change nums to [2,2,2,2]. You cannot change any number to 3 since 3 > limit.

Example 3: Input: nums = [1,2,1,2], limit = 2 Output: 0 Explanation: nums is already complementary.

Constraints:

n == nums.length 2 <= n <= 105 1 <= nums[i] <= limit <= 105 n is even.

== Problem Number: 1234 URL: https://leetcode.com/problems/max-number-of-k-sum-pairs Title: 1679. Max Number of K-Sum Pairs Problem Description: You are given an integer array nums and an integer k. In one operation, you can pick two numbers from the array whose sum equals k and remove them from the array. Return the maximum number of operations you can perform on the array. Example 1: Input: nums = [1,2,3,4], k = 5 Output: 2 Explanation: Starting with nums = [1,2,3,4]: - Remove numbers 1 and 4, then nums = [2,3] - Remove numbers 2 and 3, then nums = [] There are no more pairs that sum up to 5, hence a total of 2 operations. Example 2: Input: nums = [3,1,3,4,3], k = 6 Output: 1 Explanation: Starting with nums = [3,1,3,4,3]: - Remove the first two 3's, then nums = [1,4,3] There are no more pairs that sum up to 6, hence a total of 1 operation. Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= k <= 109

== Problem Number: 1235 URL: https://leetcode.com/problems/concatenation-of-consecutive-binary-numbers Title: 1680. Concatenation of Consecutive Binary Numbers Problem Description: Given an integer n, return the decimal value of the binary string formed by concatenating the binary representations of 1 to n in order, modulo 109 + 7. Example 1: Input: n = 1 Output: 1

Explanation: ”1” in binary corresponds to the decimal value 1.

Example 2: Input: n = 3 Output: 27 Explanation: In binary, 1, 2, and 3 corresponds to ”1”, ”10”, and ”11”. After concatenating them, we have ”11011”, which corresponds to the decimal value 27.

Example 3: Input: n = 12 Output: 505379714 Explanation: The concatenation results in ”1101110010111011110001001101010111100”. The decimal value of that is 118505380540. After modulo 109 + 7, the result is 505379714.

Constraints:

1 <= n <= 105

==
Problem Number: 1236 URL: https://leetcode.com/problems/sum-of-absolute-differences-in-a-sorted-array Title: 1685. Sum of Absolute Differences in a Sorted Array Problem Description: You are given an integer array nums sorted in non-decreasing order. Build and return an integer array result with the same length as nums such that result[i] is equal to the summation of absolute differences between nums[i] and all the other elements in the array. In other words, result[i] is equal to sum(|nums[i]-nums[j]|) where 0 <= j < nums.length and j != i (0-indexed). Example 1: Input: nums = [2,3,5] Output: [4,3,5] Explanation: Assuming the arrays are 0-indexed, then result[0] = |2-2| + |2-3| + |2-5| = 0 + 1 + 3 = 4, result[1] = |3-2| + |3-3| + |3-5| = 1 + 0 + 2 = 3, result[2] = |5-2| + |5-3| + |5-5| = 3 + 2 + 0 = 5.

Example 2: Input: nums = [1,4,6,8,10] Output: [24,15,13,15,21]

Constraints:

2 <= nums.length <= 105 1 <= nums[i] <= nums[i + 1] <= 104

==
Problem Number: 1237 URL: https://leetcode.com/problems/stone-game-vi Title: 1686. Stone Game VI Problem Description: Alice and Bob take turns playing a game, with Alice starting first. There are n stones in a pile. On each player’s turn, they can remove a stone from the pile and receive points based on the stone’s value. Alice and Bob may value the stones differently. You are given two integer arrays of length n, aliceValues and bobValues. Each aliceValues[i] and bobValues[i] represents how Alice and Bob, respectively, value the ith stone. The winner is the person with the most points after all the stones are chosen. If both players have the same amount of points, the game results in a draw. Both players will play optimally. Both players know the other’s values. Determine the result of the game, and:

If Alice wins, return 1. If Bob wins, return -1. If the game results in a draw, return 0.

Example 1: Input: aliceValues = [1,3], bobValues = [2,1] Output: 1 Explanation: If Alice takes stone 1 (0-indexed) first, Alice will receive 3 points. Bob can only choose stone 0, and will only receive 2 points. Alice wins.

Example 2: Input: aliceValues = [1,2], bobValues = [3,1] Output: 0 Explanation: If Alice takes stone 0, and Bob takes stone 1, they will both have 1 point. Draw.

Example 3: Input: aliceValues = [2,4,3], bobValues = [1,6,7] Output: -1 Explanation: Regardless of how Alice plays, Bob will be able to have more points than Alice. For example, if Alice takes stone 1, Bob can take stone 2, and Alice takes stone 0, Alice will have 6 points to Bob's 7. Bob wins.

Constraints:

n == aliceValues.length == bobValues.length 1 <= n <= 105 1 <= aliceValues[i], bobValues[i] <= 100

==

Problem Number: 1238 URL: https://leetcode.com/problems/partitioning-into-minimum-number-of-deci-binary-numbers Title: 1689. Partitioning Into Minimum Number Of Deci-Binary Numbers Problem Description: A decimal number is called deci-binary if each of its digits is either 0 or 1 without any leading zeros. For example, 101 and 1100 are deci-binary, while 112 and 3001 are not. Given a string n that represents a positive decimal integer, return the minimum number of positive deci-binary numbers needed so that they sum up to n. Example 1: Input: n = "32" Output: 3 Explanation: 10 + 11 + 11 = 32

Example 2: Input: n = "82734" Output: 8

Example 3: Input: n = "27346209830709182346" Output: 9

Constraints:

1 <= n.length <= 105 n consists of only digits. n does not contain any leading zeros and represents a positive integer.

==

Problem Number: 1239 URL: https://leetcode.com/problems/stone-game-vii Title: 1690. Stone Game VII Problem Description: Alice and Bob take turns playing a game, with Alice starting first. There are n stones arranged in a row. On each player's turn, they can remove either the leftmost stone or the rightmost stone from the row and receive points equal to the sum of the remaining stones' values in the row. The winner is the one with the higher score when there are no stones left to remove. Bob found that he will always lose this game (poor Bob, he always loses), so he decided to minimize the score's difference. Alice's goal is to maximize the difference in the score. Given an array of integers stones where stones[i] represents the value of the ith stone from the left, return the difference in Alice and Bob's score if they both play optimally. Example 1: Input: stones = [5,3,1,4,2] Output: 6 Explanation: - Alice removes 2 and gets 5 + 3 + 1 + 4 = 13 points. Alice = 13, Bob = 0, stones = [5,3,1,4]. - Bob removes 5 and gets 3 + 1 + 4 = 8 points. Alice = 13, Bob = 8, stones = [3,1,4]. - Alice removes 3 and gets 1 + 4 = 5 points. Alice

= 18, Bob = 8, stones = [1,4]. - Bob removes 1 and gets 4 points. Alice = 18, Bob = 12, stones = [4]. - Alice removes 4 and gets 0 points. Alice = 18, Bob = 12, stones = []. The score difference is 18 - 12 = 6.

Example 2: Input: stones = [7,90,5,1,100,10,10,2] Output: 122 Constraints:

n == stones.length 2 <= n <= 1000 1 <= stones[i] <= 1000

==
Problem Number: 1240 URL: https://leetcode.com/problems/maximum-erasure-value Title: 1695. Maximum Erasure Value Problem Description: You are given an array of positive integers nums and want to erase a subarray containing unique elements. The score you get by erasing the subarray is equal to the sum of its elements. Return the maximum score you can get by erasing exactly one subarray. An array b is called to be a subarray of a if it forms a contiguous subsequence of a, that is, if it is equal to a[l],a[l+1],...,a[r] for some (l,r). Example 1: Input: nums = [4,2,4,5,6] Output: 17 Explanation: The optimal subarray here is [2,4,5,6].

Example 2: Input: nums = [5,2,1,2,5,2,1,2,5] Output: 8 Explanation: The optimal subarray here is [5,2,1] or [1,2,5].

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 104

==
Problem Number: 1241 URL: https://leetcode.com/problems/jump-game-vi Title: 1696. Jump Game VI Problem Description: You are given a 0-indexed integer array nums and an integer k. You are initially standing at index 0. In one move, you can jump at most k steps forward without going outside the boundaries of the array. That is, you can jump from index i to any index in the range [i + 1, min(n - 1, i + k)] inclusive. You want to reach the last index of the array (index n - 1). Your score is the sum of all nums[j] for each index j you visited in the array. Return the maximum score you can get. Example 1: Input: nums = [1,-1,-2,4,-7,3], k = 2 Output: 7 Explanation: You can choose your jumps forming the subsequence [1,-1,4,3] (underlined above). The sum is 7.

Example 2: Input: nums = [10,-5,-2,4,0,3], k = 3 Output: 17 Explanation: You can choose your jumps forming the subsequence [10,4,3] (underlined above). The sum is 17.

Example 3: Input: nums = [1,-5,-20,4,-1,3,-6,-3], k = 2 Output: 0

Constraints:

1 <= nums.length, k <= 105 -104 <= nums[i] <= 104

==
Problem Number: 1242 URL: https://leetcode.com/problems/average-waiting-time Title: 1701. Average Waiting Time Problem Description: There is

a restaurant with a single chef. You are given an array customers, where customers[i] = [arrivali, timei]:

arrivali is the arrival time of the ith customer. The arrival times are sorted in non-decreasing order. timei is the time needed to prepare the order of the ith customer.

When a customer arrives, he gives the chef his order, and the chef starts preparing it once he is idle. The customer waits till the chef finishes preparing his order. The chef does not prepare food for more than one customer at a time. The chef prepares food for customers in the order they were given in the input. Return the average waiting time of all customers. Solutions within 10-5 from the actual answer are considered accepted. Example 1: Input: customers = [[1,2],[2,5],[4,3]] Output: 5.00000 Explanation: 1) The first customer arrives at time 1, the chef takes his order and starts preparing it immediately at time 1, and finishes at time 3, so the waiting time of the first customer is 3 - 1 = 2. 2) The second customer arrives at time 2, the chef takes his order and starts preparing it at time 3, and finishes at time 8, so the waiting time of the second customer is 8 - 2 = 6. 3) The third customer arrives at time 4, the chef takes his order and starts preparing it at time 8, and finishes at time 11, so the waiting time of the third customer is 11 - 4 = 7. So the average waiting time = (2 + 6 + 7) / 3 = 5.

Example 2: Input: customers = [[5,2],[5,4],[10,3],[20,1]] Output: 3.25000 Explanation: 1) The first customer arrives at time 5, the chef takes his order and starts preparing it immediately at time 5, and finishes at time 7, so the waiting time of the first customer is 7 - 5 = 2. 2) The second customer arrives at time 5, the chef takes his order and starts preparing it at time 7, and finishes at time 11, so the waiting time of the second customer is 11 - 5 = 6. 3) The third customer arrives at time 10, the chef takes his order and starts preparing it at time 11, and finishes at time 14, so the waiting time of the third customer is 14 - 10 = 4. 4) The fourth customer arrives at time 20, the chef takes his order and starts preparing it immediately at time 20, and finishes at time 21, so the waiting time of the fourth customer is 21 - 20 = 1. So the average waiting time = (2 + 6 + 4 + 1) / 4 = 3.25.

Constraints:

1 <= customers.length <= 105 1 <= arrivali, timei <= 104 arrivali <= arrivali+1

==
Problem Number: 1243 URL: https://leetcode.com/problems/maximum-binary-string-after-change Title: 1702. Maximum Binary String After Change Problem Description: You are given a binary string binary consisting of only 0's or 1's. You can apply each of the following operations any number of times:

Operation 1: If the number contains the substring "00", you can replace it with "10".

For example, ”00010” -> ”10010”

Operation 2: If the number contains the substring ”10”, you can replace it with ”01”.

For example, ”00010” -> ”00001”

Return the maximum binary string you can obtain after any number of operations. Binary string x is greater than binary string y if x’s decimal representation is greater than y’s decimal representation. Example 1: Input: binary = ”000110” Output: ”111011” Explanation: A valid transformation sequence can be: ”000110” -> ”000101” ”000101” -> ”100101” ”100101” -> ”110101” ”110101” -> ”110011” ”110011” -> ”111011”

Example 2: Input: binary = ”01” Output: ”01” Explanation: ”01” cannot be transformed any further.

Constraints:

1 <= binary.length <= 105 binary consist of ’0’ and ’1’.

==
Problem Number: 1244 URL: https://leetcode.com/problems/maximum-number-of-eaten-apples Title: 1705. Maximum Number of Eaten Apples Problem Description: There is a special kind of apple tree that grows apples every day for n days. On the ith day, the tree grows apples[i] apples that will rot after days[i] days, that is on day i + days[i] the apples will be rotten and cannot be eaten. On some days, the apple tree does not grow any apples, which are denoted by apples[i] == 0 and days[i] == 0. You decided to eat at most one apple a day (to keep the doctors away). Note that you can keep eating after the first n days. Given two integer arrays days and apples of length n, return the maximum number of apples you can eat. Example 1: Input: apples = [1,2,3,5,2], days = [3,2,1,4,2] Output: 7 Explanation: You can eat 7 apples: - On the first day, you eat an apple that grew on the first day. - On the second day, you eat an apple that grew on the second day. - On the third day, you eat an apple that grew on the second day. After this day, the apples that grew on the third day rot. - On the fourth to the seventh days, you eat apples that grew on the fourth day.

Example 2: Input: apples = [3,0,0,0,0,2], days = [3,0,0,0,0,2] Output: 5 Explanation: You can eat 5 apples: - On the first to the third day you eat apples that grew on the first day. - Do nothing on the fouth and fifth days. - On the sixth and seventh days you eat apples that grew on the sixth day.

Constraints:

n == apples.length == days.length 1 <= n <= 2 * 104 0 <= apples[i], days[i] <= 2 * 104 days[i] = 0 if and only if apples[i] = 0.

==
Problem Number: 1245 URL: https://leetcode.com/problems/where-will-the-

ball-fall Title: 1706. Where Will the Ball Fall Problem Description: You have a 2-D grid of size m x n representing a box, and you have n balls. The box is open on the top and bottom sides. Each cell in the box has a diagonal board spanning two corners of the cell that can redirect a ball to the right or to the left.

A board that redirects the ball to the right spans the top-left corner to the bottom-right corner and is represented in the grid as 1. A board that redirects the ball to the left spans the top-right corner to the bottom-left corner and is represented in the grid as -1.

We drop one ball at the top of each column of the box. Each ball can get stuck in the box or fall out of the bottom. A ball gets stuck if it hits a "V" shaped pattern between two boards or if a board redirects the ball into either wall of the box. Return an array answer of size n where answer[i] is the column that the ball falls out of at the bottom after dropping the ball from the ith column at the top, or -1 if the ball gets stuck in the box. Example 1:

Input: grid = [[1,1,1,-1,-1],[1,1,1,-1,-1],[-1,-1,-1,1,1],[1,1,1,1,-1],[-1,-1,-1,-1,-1]] Output: [1,-1,-1,-1,-1] Explanation: This example is shown in the photo. Ball b0 is dropped at column 0 and falls out of the box at column 1. Ball b1 is dropped at column 1 and will get stuck in the box between column 2 and 3 and row 1. Ball b2 is dropped at column 2 and will get stuck on the box between column 2 and 3 and row 0. Ball b3 is dropped at column 3 and will get stuck on the box between column 2 and 3 and row 0. Ball b4 is dropped at column 4 and will get stuck on the box between column 2 and 3 and row 1.

Example 2: Input: grid = [[-1]] Output: [-1] Explanation: The ball gets stuck against the left wall.

Example 3: Input: grid = [[1,1,1,1,1,1],[-1,-1,-1,-1,-1,-1],[1,1,1,1,1,1],[-1,-1,-1,-1,-1,-1]] Output: [0,1,2,3,4,-1]

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 100 grid[i][j] is 1 or -1.

==
Problem Number: 1246 URL: https://leetcode.com/problems/count-good-meals Title: 1711. Count Good Meals Problem Description: A good meal is a meal that contains exactly two different food items with a sum of deliciousness equal to a power of two. You can pick any two different foods to make a good meal. Given an array of integers deliciousness where deliciousness[i] is the deliciousness of the ith item of food, return the number of different good meals you can make from this list modulo 109 + 7. Note that items with different indices are considered different even if they have the same deliciousness value. Example 1: Input: deliciousness = [1,3,5,7,9] Output: 4 Explanation: The good meals are (1,3), (1,7), (3,5) and, (7,9). Their respective sums are 4, 8, 8, and 16, all of which are powers of 2.

Example 2: Input: deliciousness = [1,1,1,3,3,3,7] Output: 15 Explanation: The good meals are (1,1) with 3 ways, (1,3) with 9 ways, and (1,7) with 3 ways. Constraints:

1 <= deliciousness.length <= 105 0 <= deliciousness[i] <= 220

==

Problem Number: 1247 URL: https://leetcode.com/problems/ways-to-split-array-into-three-subarrays Title: 1712. Ways to Split Array Into Three Subarrays Problem Description: A split of an integer array is good if:

The array is split into three non-empty contiguous subarrays - named left, mid, right respectively from left to right. The sum of the elements in left is less than or equal to the sum of the elements in mid, and the sum of the elements in mid is less than or equal to the sum of the elements in right.

Given nums, an array of non-negative integers, return the number of good ways to split nums. As the number may be too large, return it modulo 109 + 7. Example 1: Input: nums = [1,1,1] Output: 1 Explanation: The only good way to split nums is [1] [1] [1]. Example 2: Input: nums = [1,2,2,2,5,0] Output: 3 Explanation: There are three good ways of splitting nums: [1] [2] [2,2,5,0] [1] [2,2] [2,5,0] [1,2] [2,2] [5,0]

Example 3: Input: nums = [3,2,1] Output: 0 Explanation: There is no good way to split nums. Constraints:

3 <= nums.length <= 105 0 <= nums[i] <= 104

==

Problem Number: 1248 URL: https://leetcode.com/problems/maximum-score-from-removing-substrings Title: 1717. Maximum Score From Removing Substrings Problem Description: You are given a string s and two integers x and y. You can perform two types of operations any number of times.

Remove substring "ab" and gain x points.

For example, when removing "ab" from "cabxbae" it becomes "cxbae".

Remove substring "ba" and gain y points.

For example, when removing "ba" from "cabxbae" it becomes "cabxe".

Return the maximum points you can gain after applying the above operations on s. Example 1: Input: s = "cdbcbbaaabab", x = 4, y = 5 Output: 19 Explanation: - Remove the "ba" underlined in "cdbcbbaaabab". Now, s = "cdbcbbaaab" and 5 points are added to the score. - Remove the "ab" underlined in "cdbcbbaaab". Now, s = "cdbcbbaa" and 4 points are added to the score. - Remove the "ba" underlined in "cdbcbbaa". Now, s = "cdbcba" and 5 points are added to the score. - Remove the "ba" underlined in "cdbcba". Now, s = "cdbc" and 5 points are added to the score. Total score = 5 + 4 + 5 + 5 = 19. Example 2: Input: s = "aabbaaxybbaabb", x = 5, y = 4 Output: 20

Constraints:

1 <= s.length <= 105 1 <= x, y <= 104 s consists of lowercase English letters.

==
Problem Number: 1249 URL: https://leetcode.com/problems/construct-the-lexicographically-largest-valid-sequence Title: 1718. Construct the Lexicographically Largest Valid Sequence Problem Description: Given an integer n, find a sequence that satisfies all of the following:

The integer 1 occurs once in the sequence. Each integer between 2 and n occurs twice in the sequence. For every integer i between 2 and n, the distance between the two occurrences of i is exactly i.

The distance between two numbers on the sequence, a[i] and a[j], is the absolute difference of their indices, |j - i|. Return the lexicographically largest sequence. It is guaranteed that under the given constraints, there is always a solution. A sequence a is lexicographically larger than a sequence b (of the same length) if in the first position where a and b differ, sequence a has a number greater than the corresponding number in b. For example, [0,1,9,0] is lexicographically larger than [0,1,5,6] because the first position they differ is at the third number, and 9 is greater than 5. Example 1: Input: n = 3 Output: [3,1,2,3,2] Explanation: [2,3,2,1,3] is also a valid sequence, but [3,1,2,3,2] is the lexicographically largest valid sequence.

Example 2: Input: n = 5 Output: [5,3,1,4,3,5,2,4,2]

Constraints:

1 <= n <= 20

==
Problem Number: 1250 URL: https://leetcode.com/problems/swapping-nodes-in-a-linked-list Title: 1721. Swapping Nodes in a Linked List Problem Description: You are given the head of a linked list, and an integer k. Return the head of the linked list after swapping the values of the kth node from the beginning and the kth node from the end (the list is 1-indexed). Example 1:

Input: head = [1,2,3,4,5], k = 2 Output: [1,4,3,2,5]

Example 2: Input: head = [7,9,6,6,7,8,3,0,9,5], k = 5 Output: [7,9,6,6,8,7,3,0,9,5]

Constraints:

The number of nodes in the list is n. 1 <= k <= n <= 105 0 <= Node.val <= 100

==
Problem Number: 1251 URL: https://leetcode.com/problems/minimize-hamming-distance-after-swap-operations Title: 1722. Minimize Hamming Distance After Swap Operations Problem Description: You are given two integer arrays, source and target, both of length n. You are also given an

array allowedSwaps where each allowedSwaps[i] = [ai, bi] indicates that you are allowed to swap the elements at index ai and index bi (0-indexed) of array source. Note that you can swap elements at a specific pair of indices multiple times and in any order. The Hamming distance of two arrays of the same length, source and target, is the number of positions where the elements are different. Formally, it is the number of indices i for 0 <= i <= n-1 where source[i] != target[i] (0-indexed). Return the minimum Hamming distance of source and target after performing any amount of swap operations on array source.   Example 1: Input: source = [1,2,3,4], target = [2,1,4,5], allowedSwaps = [[0,1],[2,3]] Output: 1 Explanation: source can be transformed the following way: - Swap indices 0 and 1: source = [2,1,3,4] - Swap indices 2 and 3: source = [2,1,4,3] The Hamming distance of source and target is 1 as they differ in 1 position: index 3.

Example 2: Input: source = [1,2,3,4], target = [1,3,2,4], allowedSwaps = [] Output: 2 Explanation: There are no allowed swaps. The Hamming distance of source and target is 2 as they differ in 2 positions: index 1 and index 2.

Example 3: Input: source = [5,1,2,4,3], target = [1,5,4,2,3], allowedSwaps = [[0,4],[4,2],[1,3],[1,4]] Output: 0

Constraints:

n == source.length == target.length 1 <= n <= 105 1 <= source[i], target[i] <= 105 0 <= allowedSwaps.length <= 105 allowedSwaps[i].length == 2 0 <= ai, bi <= n - 1 ai != bi

==

Problem Number: 1252 URL: https://leetcode.com/problems/tuple-with-same-product Title: 1726. Tuple with Same Product Problem Description: Given an array nums of distinct positive integers, return the number of tuples (a, b, c, d) such that a * b = c * d where a, b, c, and d are elements of nums, and a != b != c != d.   Example 1: Input: nums = [2,3,4,6] Output: 8 Explanation: There are 8 valid tuples: (2,6,3,4) , (2,6,4,3) , (6,2,3,4) , (6,2,4,3) (3,4,2,6) , (4,3,2,6) , (3,4,6,2) , (4,3,6,2)

Example 2: Input: nums = [1,2,4,5,10] Output: 16 Explanation: There are 16 valid tuples: (1,10,2,5) , (1,10,5,2) , (10,1,2,5) , (10,1,5,2) (2,5,1,10) , (2,5,10,1) , (5,2,1,10) , (5,2,10,1) (2,10,4,5) , (2,10,5,4) , (10,2,4,5) , (10,2,5,4) (4,5,2,10) , (4,5,10,2) , (5,4,2,10) , (5,4,10,2)

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 104 All elements in nums are distinct.

==

Problem Number: 1253 URL: https://leetcode.com/problems/largest-submatrix-with-rearrangements Title: 1727. Largest Submatrix With Rearrangements Problem Description: You are given a binary matrix matrix of size m x n, and you are allowed to rearrange the columns of the matrix in any order. Return the area of the largest submatrix within matrix where every element of the submatrix is 1 after reordering the columns optimally. Example 1:

Input: matrix = [[0,0,1],[1,1,1],[1,0,1]] Output: 4 Explanation: You can rearrange the columns as shown above. The largest submatrix of 1s, in bold, has an area of 4.

Example 2:

Input: matrix = [[1,0,1,0,1]] Output: 3 Explanation: You can rearrange the columns as shown above. The largest submatrix of 1s, in bold, has an area of 3.

Example 3: Input: matrix = [[1,1,0],[1,0,1]] Output: 2 Explanation: Notice that you must rearrange entire columns, and there is no way to make a submatrix of 1s larger than an area of 2.

Constraints:

m == matrix.length n == matrix[i].length 1 <= m * n <= 105 matrix[i][j] is either 0 or 1.

==
Problem Number: 1254 URL: https://leetcode.com/problems/minimum-number-of-people-to-teach Title: 1733. Minimum Number of People to Teach Problem Description: On a social network consisting of m users and some friendships between users, two users can communicate with each other if they know a common language. You are given an integer n, an array languages, and an array friendships where:

There are n languages numbered 1 through n, languages[i] is the set of languages the ith user knows, and friendships[i] = [ui, vi] denotes a friendship between the users ui and vi.

You can choose one language and teach it to some users so that all friends can communicate with each other. Return the minimum number of users you need to teach. Note that friendships are not transitive, meaning if x is a friend of y and y is a friend of z, this doesn't guarantee that x is a friend of z. Example 1: Input: n = 2, languages = [[1],[2],[1,2]], friendships = [[1,2],[1,3],[2,3]] Output: 1 Explanation: You can either teach user 1 the second language or user 2 the first language.

Example 2: Input: n = 3, languages = [[2],[1,3],[1,2],[3]], friendships = [[1,4],[1,2],[3,4],[2,3]] Output: 2 Explanation: Teach the third language to users 1 and 3, yielding two users to teach.

Constraints:

2 <= n <= 500 languages.length == m 1 <= m <= 500 1 <= languages[i].length <= n 1 <= languages[i][j] <= n 1 <= ui < vi <= lan-

guages.length 1 <= friendships.length <= 500 All tuples (ui, vi) are unique languages[i] contains only unique values

== Problem Number: 1255 URL: https://leetcode.com/problems/decode-xored-permutation Title: 1734. Decode XORed Permutation Problem Description: There is an integer array perm that is a permutation of the first n positive integers, where n is always odd. It was encoded into another integer array encoded of length n - 1, such that encoded[i] = perm[i] XOR perm[i + 1]. For example, if perm = [1,3,2], then encoded = [2,1]. Given the encoded array, return the original array perm. It is guaranteed that the answer exists and is unique. Example 1: Input: encoded = [3,1] Output: [1,2,3] Explanation: If perm = [1,2,3], then encoded = [1 XOR 2,2 XOR 3] = [3,1]

Example 2: Input: encoded = [6,5,4,6] Output: [2,4,1,5,3]

Constraints:

3 <= n < 105 n is odd. encoded.length == n - 1

== Problem Number: 1256 URL: https://leetcode.com/problems/change-minimum-characters-to-satisfy-one-of-three-conditions Title: 1737. Change Minimum Characters to Satisfy One of Three Conditions Problem Description: You are given two strings a and b that consist of lowercase letters. In one operation, you can change any character in a or b to any lowercase letter. Your goal is to satisfy one of the following three conditions:

Every letter in a is strictly less than every letter in b in the alphabet. Every letter in b is strictly less than every letter in a in the alphabet. Both a and b consist of only one distinct letter.

Return the minimum number of operations needed to achieve your goal. Example 1: Input: a = "aba", b = "caa" Output: 2 Explanation: Consider the best way to make each condition true: 1) Change b to "ccc" in 2 operations, then every letter in a is less than every letter in b. 2) Change a to "bbb" and b to "aaa" in 3 operations, then every letter in b is less than every letter in a. 3) Change a to "aaa" and b to "aaa" in 2 operations, then a and b consist of one distinct letter. The best way was done in 2 operations (either condition 1 or condition 3).

Example 2: Input: a = "dabadd", b = "cda" Output: 3 Explanation: The best way is to make condition 1 true by changing b to "eee".

Constraints:

1 <= a.length, b.length <= 105 a and b consist only of lowercase letters.

== Problem Number: 1257 URL: https://leetcode.com/problems/find-kth-largest-xor-coordinate-value Title: 1738. Find Kth Largest XOR Coordinate Value

Problem Description: You are given a 2D matrix of size m x n, consisting of non-negative integers. You are also given an integer k. The value of coordinate (a, b) of the matrix is the XOR of all matrix[i][j] where 0 <= i <= a < m and 0 <= j <= b < n (0-indexed). Find the kth largest value (1-indexed) of all the coordinates of matrix. Example 1: Input: matrix = [[5,2],[1,6]], k = 1 Output: 7 Explanation: The value of coordinate (0,1) is 5 XOR 2 = 7, which is the largest value.

Example 2: Input: matrix = [[5,2],[1,6]], k = 2 Output: 5 Explanation: The value of coordinate (0,0) is 5 = 5, which is the 2nd largest value.

Example 3: Input: matrix = [[5,2],[1,6]], k = 3 Output: 4 Explanation: The value of coordinate (1,0) is 5 XOR 1 = 4, which is the 3rd largest value. Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 1000 0 <= matrix[i][j] <= 106 1 <= k <= m * n

==

Problem Number: 1258 URL: https://leetcode.com/problems/restore-the-array-from-adjacent-pairs Title: 1743. Restore the Array From Adjacent Pairs Problem Description: There is an integer array nums that consists of n unique elements, but you have forgotten it. However, you do remember every pair of adjacent elements in nums. You are given a 2D integer array adjacentPairs of size n - 1 where each adjacentPairs[i] = [ui, vi] indicates that the elements ui and vi are adjacent in nums. It is guaranteed that every adjacent pair of elements nums[i] and nums[i+1] will exist in adjacentPairs, either as [nums[i], nums[i+1]] or [nums[i+1], nums[i]]. The pairs can appear in any order. Return the original array nums. If there are multiple solutions, return any of them. Example 1: Input: adjacentPairs = [[2,1],[3,4],[3,2]] Output: [1,2,3,4] Explanation: This array has all its adjacent pairs in adjacentPairs. Notice that adjacentPairs[i] may not be in left-to-right order.

Example 2: Input: adjacentPairs = [[4,-2],[1,4],[-3,1]] Output: [-2,4,1,-3] Explanation: There can be negative numbers. Another solution is [-3,1,4,-2], which would also be accepted.

Example 3: Input: adjacentPairs = [[100000,-100000]] Output: [100000,-100000]

Constraints:

nums.length == n adjacentPairs.length == n - 1 adjacentPairs[i].length == 2 2 <= n <= 105 -105 <= nums[i], ui, vi <= 105 There exists some nums that has adjacentPairs as its pairs.

==

Problem Number: 1259 URL: https://leetcode.com/problems/can-you-eat-your-favorite-candy-on-your-favorite-day Title: 1744. Can You Eat Your Favorite Candy on Your Favorite Day? Problem Description: You are given a (0-indexed) array of positive integers candiesCount where candiesCount[i] represents the number of candies of the ith type you have. You are also given a 2D array queries where queries[i] = [favoriteTypei, favoriteDayi, dailyCapi]. You play a game with the following rules:

You start eating candies on day 0. You cannot eat any candy of type i unless you have eaten all candies of type i - 1. You must eat at least one candy per day until you have eaten all the candies.

Construct a boolean array answer such that answer.length == queries.length and answer[i] is true if you can eat a candy of type favoriteTypei on day favoriteDayi without eating more than dailyCapi candies on any day, and false otherwise. Note that you can eat different types of candy on the same day, provided that you follow rule 2. Return the constructed array answer. Example 1: Input: candiesCount = [7,4,5,3,8], queries = [[0,2,2],[4,2,4],[2,13,1000000000]] Output: [true,false,true] Explanation: 1- If you eat 2 candies (type 0) on day 0 and 2 candies (type 0) on day 1, you will eat a candy of type 0 on day 2. 2- You can eat at most 4 candies each day. If you eat 4 candies every day, you will eat 4 candies (type 0) on day 0 and 4 candies (type 0 and type 1) on day 1. On day 2, you can only eat 4 candies (type 1 and type 2), so you cannot eat a candy of type 4 on day 2. 3- If you eat 1 candy each day, you will eat a candy of type 2 on day 13.

Example 2: Input: candiesCount = [5,2,6,4,1], queries = [[3,1,2],[4,10,3],[3,10,100],[4,100,30],[1,3,1]] Output: [false,true,true,false,false]

Constraints:

1 <= candiesCount.length <= 105 1 <= candiesCount[i] <= 105 1 <= queries.length <= 105 queries[i].length == 3 0 <= favoriteTypei < candiesCount.length 0 <= favoriteDayi <= 109 1 <= dailyCapi <= 109

==

Problem Number: 1260 URL: https://leetcode.com/problems/maximum-absolute-sum-of-any-subarray Title: 1749. Maximum Absolute Sum of Any Subarray Problem Description: You are given an integer array nums. The absolute sum of a subarray [numsl, numsl+1, ..., numsr-1, numsr] is abs(numsl + numsl+1 + ... + numsr-1 + numsr). Return the maximum absolute sum of any (possibly empty) subarray of nums. Note that abs(x) is defined as follows:

If x is a negative integer, then abs(x) = -x. If x is a non-negative integer, then abs(x) = x.

Example 1: Input: nums = [1,-3,2,3,-4] Output: 5 Explanation: The subarray [2,3] has absolute sum = abs(2+3) = abs(5) = 5.

Example 2: Input: nums = [2,-5,1,-4,3,-2] Output: 8 Explanation: The subarray [-5,1,-4] has absolute sum = abs(-5+1-4) = abs(-8) = 8.

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104

==
Problem Number: 1261 URL: https://leetcode.com/problems/minimum-length-of-string-after-deleting-similar-ends Title: 1750. Minimum Length of String After Deleting Similar Ends Problem Description: Given a string s consisting only of characters 'a', 'b', and 'c'. You are asked to apply the following algorithm on the string any number of times:

Pick a non-empty prefix from the string s where all the characters in the prefix are equal. Pick a non-empty suffix from the string s where all the characters in this suffix are equal. The prefix and the suffix should not intersect at any index. The characters from the prefix and suffix must be the same. Delete both the prefix and the suffix.

Return the minimum length of s after performing the above operation any number of times (possibly zero times). Example 1: Input: s = "ca" Output: 2 Explanation: You can't remove any characters, so the string stays as is.

Example 2: Input: s = "cabaabac" Output: 0 Explanation: An optimal sequence of operations is: - Take prefix = "c" and suffix = "c" and remove them, s = "abaaba". - Take prefix = "a" and suffix = "a" and remove them, s = "baab". - Take prefix = "b" and suffix = "b" and remove them, s = "aa". - Take prefix = "a" and suffix = "a" and remove them, s = "". Example 3: Input: s = "aabccabba" Output: 3 Explanation: An optimal sequence of operations is: - Take prefix = "aa" and suffix = "a" and remove them, s = "bccabb". - Take prefix = "b" and suffix = "bb" and remove them, s = "cca".

Constraints:

1 <= s.length <= 105 s only consists of characters 'a', 'b', and 'c'.

==
Problem Number: 1262 URL: https://leetcode.com/problems/maximum-score-from-removing-stones Title: 1753. Maximum Score From Removing Stones Problem Description: You are playing a solitaire game with three piles of stones of sizes a, b, and c respectively. Each turn you choose two different non-empty piles, take one stone from each, and add 1 point to your score. The game stops when there are fewer than two non-empty piles (meaning there are no more available moves). Given three integers a, b, and c, return the maximum score you can get. Example 1: Input: a = 2, b = 4, c = 6 Output: 6 Explanation: The starting state is (2, 4, 6). One optimal set of moves is: - Take from 1st and 3rd piles, state is now (1, 4, 5) - Take from 1st and 3rd piles, state is now (0, 4, 4) - Take from 2nd and 3rd piles, state is now (0, 3, 3) - Take from 2nd and 3rd piles, state is now (0, 2, 2) - Take from 2nd and 3rd piles, state is now (0, 1, 1) - Take from 2nd and 3rd piles, state is now (0, 0, 0) There are fewer than two non-empty piles, so the game ends. Total: 6 points.

Example 2: Input: a = 4, b = 4, c = 6 Output: 7 Explanation: The starting state is (4, 4, 6). One optimal set of moves is: - Take from 1st and 2nd piles, state is now (3, 3, 6) - Take from 1st and 3rd piles, state is now (2, 3, 5) - Take from 1st and 3rd piles, state is now (1, 3, 4) - Take from 1st and 3rd piles, state is now (0, 3, 3) - Take from 2nd and 3rd piles, state is now (0, 2, 2) - Take from 2nd and 3rd piles, state is now (0, 1, 1) - Take from 2nd and 3rd piles, state is now (0, 0, 0) There are fewer than two non-empty piles, so the game ends. Total: 7 points.

Example 3: Input: a = 1, b = 8, c = 8 Output: 8 Explanation: One optimal set of moves is to take from the 2nd and 3rd piles for 8 turns until they are empty. After that, there are fewer than two non-empty piles, so the game ends.

Constraints:

1 <= a, b, c <= 105

==

Problem Number: 1263 URL: https://leetcode.com/problems/largest-merge-of-two-strings Title: 1754. Largest Merge Of Two Strings Problem Description: You are given two strings word1 and word2. You want to construct a string merge in the following way: while either word1 or word2 are non-empty, choose one of the following options:

If word1 is non-empty, append the first character in word1 to merge and delete it from word1.

For example, if word1 = "abc" and merge = "dv", then after choosing this operation, word1 = "bc" and merge = "dva".

If word2 is non-empty, append the first character in word2 to merge and delete it from word2.

For example, if word2 = "abc" and merge = "", then after choosing this operation, word2 = "bc" and merge = "a".

Return the lexicographically largest merge you can construct. A string a is lexicographically larger than a string b (of the same length) if in the first position where a and b differ, a has a character strictly larger than the corresponding character in b. For example, "abcd" is lexicographically larger than "abcc" because the first position they differ is at the fourth character, and d is greater than c. Example 1: Input: word1 = "cabaa", word2 = "bcaaa" Output: "cbcabaaaaa" Explanation: One way to get the lexicographically largest merge is: - Take from word1: merge = "c", word1 = "abaa", word2 = "bcaaa" - Take from word2: merge = "cb", word1 = "abaa", word2 = "caaa" - Take from word2: merge = "cbc", word1 = "abaa", word2 = "aaa" - Take from word1: merge = "cbca", word1 = "baa", word2 = "aaa" - Take from word1: merge = "cbcab", word1 = "aa", word2 = "aaa" - Append the remaining 5 a's from word1 and word2 at the end of merge.

Example 2: Input: word1 = "abcabc", word2 = "abdcaba" Output: "abdcabcabcaba"

Constraints:

1 <= word1.length, word2.length <= 3000 word1 and word2 consist only of lowercase English letters.

==
Problem Number: 1264 URL: https://leetcode.com/problems/count-number-of-homogenous-substrings Title: 1759. Count Number of Homogenous Substrings Problem Description: Given a string s, return the number of homogenous substrings of s. Since the answer may be too large, return it modulo 109 + 7. A string is homogenous if all the characters of the string are the same. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "abbcccaa" Output: 13 Explanation: The homogenous substrings are listed as below: "a" appears 3 times. "aa" appears 1 time. "b" appears 2 times. "bb" appears 1 time. "c" appears 3 times. "cc" appears 2 times. "ccc" appears 1 time. 3 + 1 + 2 + 1 + 3 + 2 + 1 = 13. Example 2: Input: s = "xy" Output: 2 Explanation: The homogenous substrings are "x" and "y". Example 3: Input: s = "zzzzz" Output: 15

Constraints:

1 <= s.length <= 105 s consists of lowercase letters.

==
Problem Number: 1265 URL: https://leetcode.com/problems/minimum-limit-of-balls-in-a-bag Title: 1760. Minimum Limit of Balls in a Bag Problem Description: You are given an integer array nums where the ith bag contains nums[i] balls. You are also given an integer maxOperations. You can perform the following operation at most maxOperations times:

Take any bag of balls and divide it into two new bags with a positive number of balls.

For example, a bag of 5 balls can become two new bags of 1 and 4 balls, or two new bags of 2 and 3 balls.

Your penalty is the maximum number of balls in a bag. You want to minimize your penalty after the operations. Return the minimum possible penalty after performing the operations. Example 1: Input: nums = [9], maxOperations = 2 Output: 3 Explanation: - Divide the bag with 9 balls into two bags of sizes 6 and 3. [9] -> [6,3]. - Divide the bag with 6 balls into two bags of sizes 3 and 3. [6,3] -> [3,3,3]. The bag with the most number of balls has 3 balls, so your penalty is 3 and you should return 3.

Example 2: Input: nums = [2,4,8,2], maxOperations = 4 Output: 2 Explanation: - Divide the bag with 8 balls into two bags of sizes 4 and 4. [2,4,8,2] -> [2,4,4,4,2]. - Divide the bag with 4 balls into two bags of sizes 2 and 2. [2,4,4,4,2] -> [2,2,2,4,4,2]. - Divide the bag with 4 balls into two bags of sizes 2 and 2. [2,2,2,4,4,2] -> [2,2,2,2,2,4,2]. - Divide the bag with 4 balls into two bags of sizes 2 and 2. [2,2,2,2,2,4,2] -> [2,2,2,2,2,2,2,2]. The bag with the most number of balls has 2 balls, so your penalty is 2, and you should return 2.

Constraints:

1 <= nums.length <= 105 1 <= maxOperations, nums[i] <= 109

==
Problem Number: 1266 URL: https://leetcode.com/problems/form-array-by-concatenating-subarrays-of-another-array Title: 1764. Form Array by Concatenating Subarrays of Another Array Problem Description: You are given a 2D integer array groups of length n. You are also given an integer array nums. You are asked if you can choose n disjoint subarrays from the array nums such that the ith subarray is equal to groups[i] (0-indexed), and if i > 0, the (i-1)th subarray appears before the ith subarray in nums (i.e. the subarrays must be in the same order as groups). Return true if you can do this task, and false otherwise. Note that the subarrays are disjoint if and only if there is no index k such that nums[k] belongs to more than one subarray. A subarray is a contiguous sequence of elements within an array. Example 1: Input: groups = [[1,-1,-1],[3,-2,0]], nums = [1,-1,0,1,-1,-1,3,-2,0] Output: true Explanation: You can choose the 0th subarray as [1,-1,0,1,-1,-1,3,-2,0] and the 1st one as [1,-1,0,1,-1,-1,3,-2,0]. These subarrays are disjoint as they share no common nums[k] element.

Example 2: Input: groups = [[10,-2],[1,2,3,4]], nums = [1,2,3,4,10,-2] Output: false Explanation: Note that choosing the subarrays [1,2,3,4,10,-2] and [1,2,3,4,10,-2] is incorrect because they are not in the same order as in groups. [10,-2] must come before [1,2,3,4].

Example 3: Input: groups = [[1,2,3],[3,4]], nums = [7,7,1,2,3,4,7,7] Output: false Explanation: Note that choosing the subarrays [7,7,1,2,3,4,7,7] and [7,7,1,2,3,4,7,7] is invalid because they are not disjoint. They share a common elements nums[4] (0-indexed).

Constraints:

groups.length == n 1 <= n <= 103 1 <= groups[i].length, sum(groups[i].length) <= 103 1 <= nums.length <= 103 -107 <= groups[i][j], nums[k] <= 107

==
Problem Number: 1267 URL: https://leetcode.com/problems/map-of-highest-peak Title: 1765. Map of Highest Peak Problem Description: You are given an integer matrix isWater of size m x n that represents a map of land and water cells.

If isWater[i][j] == 0, cell (i, j) is a land cell. If isWater[i][j] == 1, cell (i, j) is a water cell.

You must assign each cell a height in a way that follows these rules:

The height of each cell must be non-negative. If the cell is a water cell, its height must be 0. Any two adjacent cells must have an absolute height difference of at most 1. A cell is adjacent to another cell if the former is directly north, east, south, or west of the latter (i.e., their sides are touching).

Find an assignment of heights such that the maximum height in the matrix is maximized. Return an integer matrix height of size m x n where height[i][j] is cell (i, j)'s height. If there are multiple solutions, return any of them. Example 1:

Input: isWater = [[0,1],[0,0]] Output: [[1,0],[2,1]] Explanation: The image shows the assigned heights of each cell. The blue cell is the water cell, and the green cells are the land cells.

Example 2:

Input: isWater = [[0,0,1],[1,0,0],[0,0,0]] Output: [[1,1,0],[0,1,1],[1,2,2]] Explanation: A height of 2 is the maximum possible height of any assignment. Any height assignment that has a maximum height of 2 while still meeting the rules will also be accepted.

Constraints:

m == isWater.length n == isWater[i].length 1 <= m, n <= 1000 isWater[i][j] is 0 or 1. There is at least one water cell.

==
Problem Number: 1268 URL: https://leetcode.com/problems/minimum-number-of-operations-to-move-all-balls-to-each-box Title: 1769. Minimum Number of Operations to Move All Balls to Each Box Problem Description: You have n boxes. You are given a binary string boxes of length n, where boxes[i] is '0' if the ith box is empty, and '1' if it contains one ball. In one operation, you can move one ball from a box to an adjacent box. Box i is adjacent to box j if abs(i - j) == 1. Note that after doing so, there may be more than one ball in some boxes. Return an array answer of size n, where answer[i] is the minimum number of operations needed to move all the balls to the ith box. Each answer[i] is calculated considering the initial state of the boxes. Example 1: Input: boxes = "110" Output: [1,1,3] Explanation: The answer for each box is as follows: 1) First box: you will have to move one ball from the second box to the first box in one operation. 2) Second box: you will have to move one ball from the first box to the second box in one operation. 3) Third box: you will have to move one ball from the first box to the third box in two operations, and move one ball from the second box to the third box in one operation.

Example 2: Input: boxes = "001011" Output: [11,8,5,4,3,4] Constraints:

n == boxes.length 1 <= n <= 2000 boxes[i] is either '0' or '1'.

==
Problem Number: 1269 URL: https://leetcode.com/problems/closest-dessert-cost Title: 1774. Closest Dessert Cost Problem Description: You would like to make dessert and are preparing to buy the ingredients. You have n ice cream base flavors and m types of toppings to choose from. You must follow these rules when making your dessert:

There must be exactly one ice cream base. You can add one or more types of topping or have no toppings at all. There are at most two of each type of topping.

You are given three inputs:

baseCosts, an integer array of length n, where each baseCosts[i] represents the price of the ith ice cream base flavor. toppingCosts, an integer array of length m, where each toppingCosts[i] is the price of one of the ith topping. target, an integer representing your target price for dessert.

You want to make a dessert with a total cost as close to target as possible. Return the closest possible cost of the dessert to target. If there are multiple, return the lower one.   Example 1: Input: baseCosts = [1,7], toppingCosts = [3,4], target = 10 Output: 10 Explanation: Consider the following combination (all 0-indexed): - Choose base 1: cost 7 - Take 1 of topping 0: cost 1 x 3 = 3 - Take 0 of topping 1: cost 0 x 4 = 0 Total: 7 + 3 + 0 = 10.

Example 2: Input: baseCosts = [2,3], toppingCosts = [4,5,100], target = 18 Output: 17 Explanation: Consider the following combination (all 0-indexed): - Choose base 1: cost 3 - Take 1 of topping 0: cost 1 x 4 = 4 - Take 2 of topping 1: cost 2 x 5 = 10 - Take 0 of topping 2: cost 0 x 100 = 0 Total: 3 + 4 + 10 + 0 = 17. You cannot make a dessert with a total cost of 18.

Example 3: Input: baseCosts = [3,10], toppingCosts = [2,5], target = 9 Output: 8 Explanation: It is possible to make desserts with cost 8 and 10. Return 8 as it is the lower cost.

Constraints:

n == baseCosts.length m == toppingCosts.length 1 <= n, m <= 10 1 <= baseCosts[i], toppingCosts[i] <= 104 1 <= target <= 104

==

Problem Number: 1270 URL: https://leetcode.com/problems/equal-sum-arrays-with-minimum-number-of-operations Title: 1775. Equal Sum Arrays With Minimum Number of Operations Problem Description: You are given two arrays of integers nums1 and nums2, possibly of different lengths. The values in the arrays are between 1 and 6, inclusive. In one operation, you can change any integer's value in any of the arrays to any value between 1 and 6, inclusive. Return the minimum number of operations required to make the sum of values in nums1 equal to the sum of values in nums2. Return -1 if it is not possible to make the sum of the two arrays equal.   Example 1: Input: nums1 = [1,2,3,4,5,6], nums2 = [1,1,2,2,2,2] Output: 3 Explanation: You can make the sums of nums1 and nums2 equal with 3 operations. All indices are 0-indexed. - Change nums2[0] to 6. nums1 = [1,2,3,4,5,6], nums2 = [6,1,2,2,2,2]. - Change nums1[5] to 1. nums1 = [1,2,3,4,5,1], nums2 = [6,1,2,2,2,2]. - Change nums1[2] to 2. nums1 = [1,2,2,4,5,1], nums2 = [6,1,2,2,2,2].

Example 2: Input: nums1 = [1,1,1,1,1,1,1], nums2 = [6] Output: -1 Explanation:

There is no way to decrease the sum of nums1 or to increase the sum of nums2 to make them equal.

Example 3: Input: nums1 = [6,6], nums2 = [1] Output: 3 Explanation: You can make the sums of nums1 and nums2 equal with 3 operations. All indices are 0-indexed. - Change nums1[0] to 2. nums1 = [2,6], nums2 = [1]. - Change nums1[1] to 2. nums1 = [2,2], nums2 = [1]. - Change nums2[0] to 4. nums1 = [2,2], nums2 = [4].

Constraints:

1 <= nums1.length, nums2.length <= 105 1 <= nums1[i], nums2[i] <= 6

==
Problem Number: 1271 URL: https://leetcode.com/problems/check-if-number-is-a-sum-of-powers-of-three Title: 1780. Check if Number is a Sum of Powers of Three Problem Description: Given an integer n, return true if it is possible to represent n as the sum of distinct powers of three. Otherwise, return false. An integer y is a power of three if there exists an integer x such that y == 3x. Example 1: Input: n = 12 Output: true Explanation: 12 = 31 + 32

Example 2: Input: n = 91 Output: true Explanation: 91 = 30 + 32 + 34

Example 3: Input: n = 21 Output: false

Constraints:

1 <= n <= 107

==
Problem Number: 1272 URL: https://leetcode.com/problems/sum-of-beauty-of-all-substrings Title: 1781. Sum of Beauty of All Substrings Problem Description: The beauty of a string is the difference in frequencies between the most frequent and least frequent characters.

For example, the beauty of "abaacc" is 3 - 1 = 2.

Given a string s, return the sum of beauty of all of its substrings. Example 1: Input: s = "aabcb" Output: 5 Explanation: The substrings with non-zero beauty are ["aab","aabc","aabcb","abcb","bcb"], each with beauty equal to 1. Example 2: Input: s = "aabcbaa" Output: 17

Constraints:

1 <= s.length <= 500 s consists of only lowercase English letters.

==
Problem Number: 1273 URL: https://leetcode.com/problems/minimum-elements-to-add-to-form-a-given-sum Title: 1785. Minimum Elements to Add to Form a Given Sum Problem Description: You are given an integer array nums and two integers limit and goal. The array nums has an interesting property that abs(nums[i]) <= limit. Return the minimum number of elements you need to add to make the sum of the array equal to goal. The array must

maintain its property that abs(nums[i]) <= limit. Note that abs(x) equals x if x >= 0, and -x otherwise. Example 1: Input: nums = [1,-1,1], limit = 3, goal = -4 Output: 2 Explanation: You can add -2 and -3, then the sum of the array will be 1 - 1 + 1 - 2 - 3 = -4.

Example 2: Input: nums = [1,-10,9,1], limit = 100, goal = 0 Output: 1

Constraints:

1 <= nums.length <= 105 1 <= limit <= 106 -limit <= nums[i] <= limit -109 <= goal <= 109

==
Problem Number: 1274 URL: https://leetcode.com/problems/number-of-restricted-paths-from-first-to-last-node Title: 1786. Number of Restricted Paths From First to Last Node Problem Description: There is an undirected weighted connected graph. You are given a positive integer n which denotes that the graph has n nodes labeled from 1 to n, and an array edges where each edges[i] = [ui, vi, weighti] denotes that there is an edge between nodes ui and vi with weight equal to weighti. A path from node start to node end is a sequence of nodes [z0, z1, z2, ..., zk] such that z0 = start and zk = end and there is an edge between zi and zi+1 where 0 <= i <= k-1. The distance of a path is the sum of the weights on the edges of the path. Let distanceToLastNode(x) denote the shortest distance of a path between node n and node x. A restricted path is a path that also satisfies that distanceToLastNode(zi) > distanceToLastNode(zi+1) where 0 <= i <= k-1. Return the number of restricted paths from node 1 to node n. Since that number may be too large, return it modulo 109 + 7. Example 1:

Input: n = 5, edges = [[1,2,3],[1,3,3],[2,3,1],[1,4,2],[5,2,2],[3,5,1],[5,4,10]] Output: 3 Explanation: Each circle contains the node number in black and its distanceToLastNode value in blue. The three restricted paths are: 1) 1 --> 2 --> 5 2) 1 --> 2 --> 3 --> 5 3) 1 --> 3 --> 5

Example 2:

Input: n = 7, edges = [[1,3,1],[4,1,2],[7,3,4],[2,5,3],[5,6,1],[6,7,2],[7,5,3],[2,6,4]] Output: 1 Explanation: Each circle contains the node number in black and its distanceToLastNode value in blue. The only restricted path is 1 --> 3 --> 7.

Constraints:

1 <= n <= 2 * 104 n - 1 <= edges.length <= 4 * 104 edges[i].length == 3 1 <= ui, vi <= n ui != vi 1 <= weighti <= 105 There is at most one edge between any two nodes. There is at least one path between any two nodes.

==
Problem Number: 1275 URL: https://leetcode.com/problems/maximum-average-pass-ratio Title: 1792. Maximum Average Pass Ratio Problem Description: There is a school that has classes of students and each class will be having a final exam. You are given a 2D integer array classes, where

classes[i] = [passi, totali]. You know beforehand that in the ith class, there are totali total students, but only passi number of students will pass the exam. You are also given an integer extraStudents. There are another extraStudents brilliant students that are guaranteed to pass the exam of any class they are assigned to. You want to assign each of the extraStudents students to a class in a way that maximizes the average pass ratio across all the classes. The pass ratio of a class is equal to the number of students of the class that will pass the exam divided by the total number of students of the class. The average pass ratio is the sum of pass ratios of all the classes divided by the number of the classes. Return the maximum possible average pass ratio after assigning the extraStudents students. Answers within 10-5 of the actual answer will be accepted.   Example 1: Input: classes = [[1,2],[3,5],[2,2]], extraStudents = 2 Output: 0.78333 Explanation: You can assign the two extra students to the first class. The average pass ratio will be equal to (3/4 + 3/5 + 2/2) / 3 = 0.78333.

Example 2: Input: classes = [[2,4],[3,9],[4,5],[2,10]], extraStudents = 4 Output: 0.53485

Constraints:

1 <= classes.length <= 105 classes[i].length == 2 1 <= passi <= totali <= 105 1 <= extraStudents <= 105

==
Problem Number: 1276 URL: https://leetcode.com/problems/design-authentication-manager Title: 1797. Design Authentication Manager Problem Description: There is an authentication system that works with authentication tokens. For each session, the user will receive a new authentication token that will expire timeToLive seconds after the currentTime. If the token is renewed, the expiry time will be extended to expire timeToLive seconds after the (potentially different) currentTime. Implement the AuthenticationManager class:

AuthenticationManager(int timeToLive) constructs the AuthenticationManager and sets the timeToLive. generate(string tokenId, int currentTime) generates a new token with the given tokenId at the given currentTime in seconds. renew(string tokenId, int currentTime) renews the unexpired token with the given tokenId at the given currentTime in seconds. If there are no unexpired tokens with the given tokenId, the request is ignored, and nothing happens. countUnexpiredTokens(int currentTime) returns the number of unexpired tokens at the given currentTime.

Note that if a token expires at time t, and another action happens on time t (renew or countUnexpiredTokens), the expiration takes place before the other actions.   Example 1:

Input ["AuthenticationManager", "renew", "generate", "countUnexpiredTokens", "generate", "renew", "renew", "countUnexpiredTokens"] [[5], ["aaa", 1], ["aaa", 2], [6], ["bbb", 7], ["aaa", 8], ["bbb", 10], [15]] Output [null, null, null, 1, null, null, null, 0]

Explanation AuthenticationManager authenticationManager = new AuthenticationManager(5); // Constructs the AuthenticationManager with timeToLive = 5 seconds. authenticationManager.renew("aaa", 1); // No token exists with tokenId "aaa" at time 1, so nothing happens. authenticationManager.generate("aaa", 2); // Generates a new token with tokenId "aaa" at time 2. authenticationManager.countUnexpiredTokens(6); // The token with tokenId "aaa" is the only unexpired one at time 6, so return 1. authenticationManager.generate("bbb", 7); // Generates a new token with tokenId "bbb" at time 7. authenticationManager.renew("aaa", 8); // The token with tokenId "aaa" expired at time 7, and 8 >= 7, so at time 8 the renew request is ignored, and nothing happens. authenticationManager.renew("bbb", 10); // The token with tokenId "bbb" is unexpired at time 10, so the renew request is fulfilled and now the token will expire at time 15. authenticationManager.countUnexpiredTokens(15); // The token with tokenId "bbb" expires at time 15, and the token with tokenId "aaa" expired at time 7, so currently no token is unexpired, so return 0.

Constraints:

1 <= timeToLive <= 108 1 <= currentTime <= 108 1 <= tokenId.length <= 5 tokenId consists only of lowercase letters. All calls to generate will contain unique values of tokenId. The values of currentTime across all the function calls will be strictly increasing. At most 2000 calls will be made to all functions combined.

==
Problem Number: 1277 URL: https://leetcode.com/problems/maximum-number-of-consecutive-values-you-can-make Title: 1798. Maximum Number of Consecutive Values You Can Make Problem Description: You are given an integer array coins of length n which represents the n coins that you own. The value of the ith coin is coins[i]. You can make some value x if you can choose some of your n coins such that their values sum up to x. Return the maximum number of consecutive integer values that you can make with your coins starting from and including 0. Note that you may have multiple coins of the same value. Example 1: Input: coins = [1,3] Output: 2 Explanation: You can make the following values: - 0: take [] - 1: take [1] You can make 2 consecutive integer values starting from 0. Example 2: Input: coins = [1,1,1,4] Output: 8 Explanation: You can make the following values: - 0: take [] - 1: take [1] - 2: take [1,1] - 3: take [1,1,1] - 4: take [4] - 5: take [4,1] - 6: take [4,1,1] - 7: take [4,1,1,1] You can make 8 consecutive integer values starting from 0. Example 3: Input: nums = [1,4,10,3,1] Output: 20 Constraints:

coins.length == n 1 <= n <= 4 * 104 1 <= coins[i] <= 4 * 104

==
Problem Number: 1278 URL: https://leetcode.com/problems/number-of-

orders-in-the-backlog Title: 1801. Number of Orders in the Backlog Problem Description: You are given a 2D integer array orders, where each orders[i] = [pricei, amounti, orderTypei] denotes that amounti orders have been placed of type orderTypei at the price pricei. The orderTypei is:

0 if it is a batch of buy orders, or 1 if it is a batch of sell orders.

Note that orders[i] represents a batch of amounti independent orders with the same price and order type. All orders represented by orders[i] will be placed before all orders represented by orders[i+1] for all valid i. There is a backlog that consists of orders that have not been executed. The backlog is initially empty. When an order is placed, the following happens:

If the order is a buy order, you look at the sell order with the smallest price in the backlog. If that sell order's price is smaller than or equal to the current buy order's price, they will match and be executed, and that sell order will be removed from the backlog. Else, the buy order is added to the backlog. Vice versa, if the order is a sell order, you look at the buy order with the largest price in the backlog. If that buy order's price is larger than or equal to the current sell order's price, they will match and be executed, and that buy order will be removed from the backlog. Else, the sell order is added to the backlog.

Return the total amount of orders in the backlog after placing all the orders from the input. Since this number can be large, return it modulo 109 + 7. Example 1:

Input: orders = [[10,5,0],[15,2,1],[25,1,1],[30,4,0]] Output: 6 Explanation: Here is what happens with the orders: - 5 orders of type buy with price 10 are placed. There are no sell orders, so the 5 orders are added to the backlog. - 2 orders of type sell with price 15 are placed. There are no buy orders with prices larger than or equal to 15, so the 2 orders are added to the backlog. - 1 order of type sell with price 25 is placed. There are no buy orders with prices larger than or equal to 25 in the backlog, so this order is added to the backlog. - 4 orders of type buy with price 30 are placed. The first 2 orders are matched with the 2 sell orders of the least price, which is 15 and these 2 sell orders are removed from the backlog. The 3rd order is matched with the sell order of the least price, which is 25 and this sell order is removed from the backlog. Then, there are no more sell orders in the backlog, so the 4th order is added to the backlog. Finally, the backlog has 5 buy orders with price 10, and 1 buy order with price 30. So the total number of orders in the backlog is 6.

Example 2:

Input: orders = [[7,1000000000,1],[15,3,0],[5,999999995,0],[5,1,1]] Output: 999999984 Explanation: Here is what happens with the orders: - 109 orders of type sell with price 7 are placed. There are no buy orders, so the 109 orders are added to the backlog. - 3 orders of type buy with price 15 are placed. They are matched with the 3 sell orders with the least price which is 7, and those 3 sell orders are removed from the backlog. - 999999995 orders of type buy with price 5 are placed. The least price of a sell order is 7, so the 999999995 orders are added to the backlog. - 1 order of type sell with price 5 is placed. It is matched with the buy order of the highest price, which is 5, and that buy order is removed from the backlog. Finally, the backlog has (1000000000-3) sell orders with price 7, and (999999995-1) buy orders with price 5. So the total number of orders = 1999999991, which is equal to 999999984 % (109 + 7).

Constraints:

1 <= orders.length <= 105 orders[i].length == 3 1 <= pricei, amounti <= 109 orderTypei is either 0 or 1.

==

Problem Number: 1279 URL: https://leetcode.com/problems/maximum-value-at-a-given-index-in-a-bounded-array Title: 1802. Maximum Value at a Given Index in a Bounded Array Problem Description: You are given three positive integers: n, index, and maxSum. You want to construct an array nums (0-indexed) that satisfies the following conditions:

nums.length == n nums[i] is a positive integer where 0 <= i < n. abs(nums[i] - nums[i+1]) <= 1 where 0 <= i < n-1. The sum of all the elements of nums does not exceed maxSum. nums[index] is maximized.

Return nums[index] of the constructed array. Note that abs(x) equals x if x >= 0, and -x otherwise. Example 1: Input: n = 4, index = 2, maxSum = 6 Output: 2 Explanation: nums = [1,2,2,1] is one array that satisfies all the conditions. There are no arrays that satisfy all the conditions and have nums[2] == 3, so 2 is the maximum nums[2].

Example 2: Input: n = 6, index = 1, maxSum = 10 Output: 3

Constraints:

1 <= n <= maxSum <= 109 0 <= index < n

==

Problem Number: 1280 URL: https://leetcode.com/problems/minimum-number-of-operations-to-reinitialize-a-permutation Title: 1806. Minimum Number of Operations to Reinitialize a Permutation Problem Description: You are given an even integer n. You initially have a permutation perm of size n where perm[i] == i (0-indexed). In one operation, you will create a new array arr, and for each i:

If i % 2 == 0, then arr[i] = perm[i / 2]. If i % 2 == 1, then arr[i] = perm[n / 2 + (i - 1) / 2].

You will then assign arr to perm. Return the minimum non-zero number of operations you need to perform on perm to return the permutation to its initial value. Example 1: Input: n = 2 Output: 1 Explanation: perm = [0,1] initially. After the 1st operation, perm = [0,1] So it takes only 1 operation.

Example 2: Input: n = 4 Output: 2 Explanation: perm = [0,1,2,3] initially. After the 1st operation, perm = [0,2,1,3] After the 2nd operation, perm = [0,1,2,3] So it takes only 2 operations.

Example 3: Input: n = 6 Output: 4

Constraints:

2 <= n <= 1000 n is even.

==
Problem Number: 1281 URL: https://leetcode.com/problems/evaluate-the-bracket-pairs-of-a-string Title: 1807. Evaluate the Bracket Pairs of a String Problem Description: You are given a string s that contains some bracket pairs, with each pair containing a non-empty key.

For example, in the string "(name)is(age)yearsold", there are two bracket pairs that contain the keys "name" and "age".

You know the values of a wide range of keys. This is represented by a 2D string array knowledge where each knowledge[i] = [keyi, valuei] indicates that key keyi has a value of valuei. You are tasked to evaluate all of the bracket pairs. When you evaluate a bracket pair that contains some key keyi, you will:

Replace keyi and the bracket pair with the key's corresponding valuei. If you do not know the value of the key, you will replace keyi and the bracket pair with a question mark "?" (without the quotation marks).

Each key will appear at most once in your knowledge. There will not be any nested brackets in s. Return the resulting string after evaluating all of the bracket pairs. Example 1: Input: s = "(name)is(age)yearsold", knowledge = [["name","bob"],["age","two"]] Output: "bobistwoyearsold" Explanation: The key "name" has a value of "bob", so replace "(name)" with "bob". The key "age" has a value of "two", so replace "(age)" with "two".

Example 2: Input: s = "hi(name)", knowledge = [["a","b"]] Output: "hi?" Explanation: As you do not know the value of the key "name", replace "(name)" with "?".

Example 3: Input: s = "(a)(a)(a)aaa", knowledge = [["a","yes"]] Output: "yesyesyesaaa" Explanation: The same key can appear multiple times. The key "a" has a value of "yes", so replace all occurrences of "(a)" with "yes". Notice that the "a"s not in a bracket pair are not evaluated.

Constraints:

1 <= s.length <= 105 0 <= knowledge.length <= 105 knowledge[i].length == 2 1 <= keyi.length, valuei.length <= 10 s consists of lowercase English letters and round brackets '(' and ')'. Every open bracket '(' in s will have a corresponding close bracket ')'. The key in each bracket pair of s will be non-empty. There will not be any nested bracket pairs in s. keyi and valuei consist of lowercase English letters. Each keyi in knowledge is unique.

==
Problem Number: 1282 URL: https://leetcode.com/problems/sentence-similarity-iii Title: 1813. Sentence Similarity III Problem Description: A sentence is a list of words that are separated by a single space with no leading or trailing spaces. For example, "Hello World", "HELLO", "hello world hello world" are all sentences. Words consist of only uppercase and lowercase English letters. Two sentences sentence1 and sentence2 are similar if it is possible to insert an arbitrary sentence (possibly empty) inside one of these sentences such that the two sentences become equal. For example, sentence1 = "Hello my name is Jane" and sentence2 = "Hello Jane" can be made equal by inserting "my name is" between "Hello" and "Jane" in sentence2. Given two sentences sentence1 and sentence2, return true if sentence1 and sentence2 are similar. Otherwise, return false. Example 1: Input: sentence1 = "My name is Haley", sentence2 = "My Haley" Output: true Explanation: sentence2 can be turned to sentence1 by inserting "name is" between "My" and "Haley".

Example 2: Input: sentence1 = "of", sentence2 = "A lot of words" Output: false Explanation: No single sentence can be inserted inside one of the sentences to make it equal to the other.

Example 3: Input: sentence1 = "Eating right now", sentence2 = "Eating" Output: true Explanation: sentence2 can be turned to sentence1 by inserting "right now" at the end of the sentence.

Constraints:

1 <= sentence1.length, sentence2.length <= 100 sentence1 and sentence2 consist of lowercase and uppercase English letters and spaces. The words in sentence1 and sentence2 are separated by a single space.

==
Problem Number: 1283 URL: https://leetcode.com/problems/count-nice-pairs-in-an-array Title: 1814. Count Nice Pairs in an Array Problem Description: You are given an array nums that consists of non-negative integers. Let us define rev(x) as the reverse of the non-negative integer x. For example, rev(123) = 321, and rev(120) = 21. A pair of indices (i, j) is nice if it satisfies all of the following conditions:

0 <= i < j < nums.length nums[i] + rev(nums[j]) == nums[j] + rev(nums[i])

Return the number of nice pairs of indices. Since that number can be too large, return it modulo 109 + 7. Example 1: Input: nums = [42,11,1,97] Output: 2 Explanation: The two pairs are: - (0,3) : 42 + rev(97) = 42 + 79 = 121, 97 + rev(42) = 97 + 24 = 121. - (1,2) : 11 + rev(1) = 11 + 1 = 12, 1 + rev(11) = 1 + 11 = 12.

Example 2: Input: nums = [13,10,35,24,76] Output: 4

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==
Problem Number: 1284 URL: https://leetcode.com/problems/finding-the-users-active-minutes Title: 1817. Finding the Users Active Minutes Problem Description: You are given the logs for users' actions on LeetCode, and an integer k. The logs are represented by a 2D integer array logs where each logs[i] = [IDi, timei] indicates that the user with IDi performed an action at the minute timei. Multiple users can perform actions simultaneously, and a single user can perform multiple actions in the same minute. The user active minutes (UAM) for a given user is defined as the number of unique minutes in which the user performed an action on LeetCode. A minute can only be counted once, even if multiple actions occur during it. You are to calculate a 1-indexed array answer of size k such that, for each j (1 <= j <= k), answer[j] is the number of users whose UAM equals j. Return the array answer as described above. Example 1: Input: logs = [[0,5],[1,2],[0,2],[0,5],[1,3]], k = 5 Output: [0,2,0,0,0] Explanation: The user with ID=0 performed actions at minutes 5, 2, and 5 again. Hence, they have a UAM of 2 (minute 5 is only counted once). The user with ID=1 performed actions at minutes 2 and 3. Hence, they have a UAM of 2. Since both users have a UAM of 2, answer[2] is 2, and the remaining answer[j] values are 0.

Example 2: Input: logs = [[1,1],[2,2],[2,3]], k = 4 Output: [1,1,0,0] Explanation: The user with ID=1 performed a single action at minute 1. Hence, they have a UAM of 1. The user with ID=2 performed actions at minutes 2 and 3. Hence, they have a UAM of 2. There is one user with a UAM of 1 and one with a UAM of 2. Hence, answer[1] = 1, answer[2] = 1, and the remaining values are 0.

Constraints:

1 <= logs.length <= 104 0 <= IDi <= 109 1 <= timei <= 105 k is in the range [The maximum UAM for a user, 105].

==
Problem Number: 1285 URL: https://leetcode.com/problems/minimum-absolute-sum-difference Title: 1818. Minimum Absolute Sum Difference Problem Description: You are given two positive integer arrays nums1 and nums2, both of length n. The absolute sum difference of arrays nums1 and nums2 is defined as the sum of |nums1[i] - nums2[i]| for each 0 <= i < n (0-indexed). You can replace at most one element of nums1 with any other element in nums1 to minimize the absolute sum difference. Return the minimum absolute sum difference after replacing at most one element in the array nums1. Since the answer may be large, return it modulo 109 + 7. |x| is defined as:

x if x >= 0, or -x if x < 0.

Example 1: Input: nums1 = [1,7,5], nums2 = [2,3,5] Output: 3 Explanation: There are two possible optimal solutions: - Replace the second element with the first: [1,7,5] => [1,1,5], or - Replace the second element with the third: [1,7,5]

=> [1,5,5]. Both will yield an absolute sum difference of |1-2| + (|1-3| or |5-3|) + |5-5| = 3.

Example 2: Input: nums1 = [2,4,6,8,10], nums2 = [2,4,6,8,10] Output: 0 Explanation: nums1 is equal to nums2 so no replacement is needed. This will result in an absolute sum difference of 0.

Example 3: Input: nums1 = [1,10,4,4,2,7], nums2 = [9,3,5,1,7,4] Output: 20 Explanation: Replace the first element with the second: [1,10,4,4,2,7] => [10,10,4,4,2,7]. This yields an absolute sum difference of |10-9| + |10-3| + |4-5| + |4-1| + |2-7| + |7-4| = 20

Constraints:

n == nums1.length n == nums2.length 1 <= n <= 105 1 <= nums1[i], nums2[i] <= 105

==

Problem Number: 1286 URL: https://leetcode.com/problems/find-the-winner-of-the-circular-game Title: 1823. Find the Winner of the Circular Game Problem Description: There are n friends that are playing a game. The friends are sitting in a circle and are numbered from 1 to n in clockwise order. More formally, moving clockwise from the ith friend brings you to the (i+1)th friend for 1 <= i < n, and moving clockwise from the nth friend brings you to the 1st friend. The rules of the game are as follows:

Start at the 1st friend. Count the next k friends in the clockwise direction including the friend you started at. The counting wraps around the circle and may count some friends more than once. The last friend you counted leaves the circle and loses the game. If there is still more than one friend in the circle, go back to step 2 starting from the friend immediately clockwise of the friend who just lost and repeat. Else, the last friend in the circle wins the game.

Given the number of friends, n, and an integer k, return the winner of the game. Example 1:

Input: n = 5, k = 2 Output: 3 Explanation: Here are the steps of the game: 1) Start at friend 1. 2) Count 2 friends clockwise, which are friends 1 and 2. 3) Friend 2 leaves the circle. Next start is friend 3. 4) Count 2 friends clockwise, which are friends 3 and 4. 5) Friend 4 leaves the circle. Next start is friend 5. 6) Count 2 friends clockwise, which are friends 5 and 1. 7) Friend 1 leaves the circle. Next start is friend 3. 8) Count 2 friends clockwise, which are friends 3 and 5. 9) Friend 5 leaves the circle. Only friend 3 is left, so they are the winner. Example 2: Input: n = 6, k = 5 Output: 1 Explanation: The friends leave in this order: 5, 4, 6, 2, 3. The winner is friend 1.

Constraints:

1 <= k <= n <= 500

Follow up: Could you solve this problem in linear time with constant space?

==
Problem Number: 1287 URL: https://leetcode.com/problems/minimum-sideway-jumps Title: 1824. Minimum Sideway Jumps Problem Description: There is a 3 lane road of length n that consists of n + 1 points labeled from 0 to n. A frog starts at point 0 in the second lane and wants to jump to point n. However, there could be obstacles along the way. You are given an array obstacles of length n + 1 where each obstacles[i] (ranging from 0 to 3) describes an obstacle on the lane obstacles[i] at point i. If obstacles[i] == 0, there are no obstacles at point i. There will be at most one obstacle in the 3 lanes at each point.

For example, if obstacles[2] == 1, then there is an obstacle on lane 1 at point 2.

The frog can only travel from point i to point i + 1 on the same lane if there is not an obstacle on the lane at point i + 1. To avoid obstacles, the frog can also perform a side jump to jump to another lane (even if they are not adjacent) at the same point if there is no obstacle on the new lane.

For example, the frog can jump from lane 3 at point 3 to lane 1 at point 3.

Return the minimum number of side jumps the frog needs to reach any lane at point n starting from lane 2 at point 0. Note: There will be no obstacles on points 0 and n. Example 1:

Input: obstacles = [0,1,2,3,0] Output: 2 Explanation: The optimal solution is shown by the arrows above. There are 2 side jumps (red arrows). Note that the frog can jump over obstacles only when making side jumps (as shown at point 2).

Example 2:

Input: obstacles = [0,1,1,3,3,0] Output: 0 Explanation: There are no obstacles on lane 2. No side jumps are required.

Example 3:

Input: obstacles = [0,2,1,0,3,0] Output: 2 Explanation: The optimal solution is shown by the arrows above. There are 2 side jumps.

Constraints:

obstacles.length == n + 1 1 <= n <= 5 * 105 0 <= obstacles[i] <= 3 obstacles[0] == obstacles[n] == 0

==
Problem Number: 1288 URL: https://leetcode.com/problems/queries-on-number-of-points-inside-a-circle Title: 1828. Queries on Number of Points Inside a Circle Problem Description: You are given an array points where points[i] = [xi, yi] is the coordinates of the ith point on a 2D plane. Multiple points can have the same coordinates. You are also given an array queries where queries[j] = [xj, yj, rj] describes a circle centered at (xj, yj) with a radius

of rj. For each query queries[j], compute the number of points inside the jth circle. Points on the border of the circle are considered inside. Return an array answer, where answer[j] is the answer to the jth query.   Example 1:

Input: points = [[1,3],[3,3],[5,3],[2,2]], queries = [[2,3,1],[4,3,1],[1,1,2]] Output: [3,2,2] Explanation: The points and circles are shown above. queries[0] is the green circle, queries[1] is the red circle, and queries[2] is the blue circle.

Example 2:

Input: points = [[1,1],[2,2],[3,3],[4,4],[5,5]], queries = [[1,2,2],[2,2,2],[4,3,2],[4,3,3]] Output: [2,3,2,4] Explanation: The points and circles are shown above. queries[0] is green, queries[1] is red, queries[2] is blue, and queries[3] is purple.

Constraints:

1 <= points.length <= 500 points[i].length == 2 0 <= xi, yi <= 500 1 <= queries.length <= 500 queries[j].length == 3 0 <= xj, yj <= 500 1 <= rj <= 500 All coordinates are integers.

Follow up: Could you find the answer for each query in better complexity than O(n)?

==
Problem Number: 1289 URL: https://leetcode.com/problems/maximum-xor-for-each-query Title: 1829. Maximum XOR for Each Query Problem Description: You are given a sorted array nums of n non-negative integers and an integer maximumBit. You want to perform the following query n times:

Find a non-negative integer k < 2maximumBit such that nums[0] XOR nums[1] XOR ... XOR nums[nums.length-1] XOR k is maximized. k is the answer to the ith query. Remove the last element from the current array nums.

Return an array answer, where answer[i] is the answer to the ith query.   Example 1: Input: nums = [0,1,1,3], maximumBit = 2 Output: [0,3,2,3] Explanation: The queries are answered as follows: 1st query: nums = [0,1,1,3], k = 0 since 0 XOR 1 XOR 1 XOR 3 XOR 0 = 3. 2nd query: nums = [0,1,1], k = 3 since 0 XOR 1 XOR 1 XOR 3 = 3. 3rd query: nums = [0,1], k = 2 since 0 XOR 1 XOR 2 = 3. 4th query: nums = [0], k = 3 since 0 XOR 3 = 3.

Example 2: Input: nums = [2,3,4,7], maximumBit = 3 Output: [5,2,6,5] Explanation: The queries are answered as follows: 1st query: nums = [2,3,4,7], k = 5 since 2 XOR 3 XOR 4 XOR 7 XOR 5 = 7. 2nd query: nums = [2,3,4], k = 2 since 2 XOR 3 XOR 4 XOR 2 = 7. 3rd query: nums = [2,3], k = 6 since 2 XOR 3 XOR 6 = 7. 4th query: nums = [2], k = 5 since 2 XOR 5 = 7.

Example 3: Input: nums = [0,1,2,2,5,7], maximumBit = 3 Output: [4,3,6,4,6,7]

Constraints:

nums.length == n 1 <= n <= 105 1 <= maximumBit <= 20 0 <= nums[i] < 2maximumBit nums is sorted in ascending order.

== Problem Number: 1290 URL: https://leetcode.com/problems/maximum-ice-cream-bars Title: 1833. Maximum Ice Cream Bars Problem Description: It is a sweltering summer day, and a boy wants to buy some ice cream bars. At the store, there are n ice cream bars. You are given an array costs of length n, where costs[i] is the price of the ith ice cream bar in coins. The boy initially has coins coins to spend, and he wants to buy as many ice cream bars as possible. Note: The boy can buy the ice cream bars in any order. Return the maximum number of ice cream bars the boy can buy with coins coins. You must solve the problem by counting sort. Example 1: Input: costs = [1,3,2,4,1], coins = 7 Output: 4 Explanation: The boy can buy ice cream bars at indices 0,1,2,4 for a total price of 1 + 3 + 2 + 1 = 7.

Example 2: Input: costs = [10,6,8,7,7,8], coins = 5 Output: 0 Explanation: The boy cannot afford any of the ice cream bars.

Example 3: Input: costs = [1,6,3,1,2,5], coins = 20 Output: 6 Explanation: The boy can buy all the ice cream bars for a total price of 1 + 6 + 3 + 1 + 2 + 5 = 18.

Constraints:

costs.length == n 1 <= n <= 105 1 <= costs[i] <= 105 1 <= coins <= 108

== Problem Number: 1291 URL: https://leetcode.com/problems/single-threaded-cpu Title: 1834. Single-Threaded CPU Problem Description: You are given n tasks labeled from 0 to n - 1 represented by a 2D integer array tasks, where tasks[i] = [enqueueTimei, processingTimei] means that the ith task will be available to process at enqueueTimei and will take processingTimei to finish processing. You have a single-threaded CPU that can process at most one task at a time and will act in the following way:

If the CPU is idle and there are no available tasks to process, the CPU remains idle. If the CPU is idle and there are available tasks, the CPU will choose the one with the shortest processing time. If multiple tasks have the same shortest processing time, it will choose the task with the smallest index. Once a task is started, the CPU will process the entire task without stopping. The CPU can finish a task then start a new one instantly.

Return the order in which the CPU will process the tasks. Example 1: Input: tasks = [[1,2],[2,4],[3,2],[4,1]] Output: [0,2,3,1] Explanation: The events go as follows: - At time = 1, task 0 is available to process. Available tasks = {0}. - Also at time = 1, the idle CPU starts processing task 0. Available tasks = {}. - At time = 2, task 1 is available to process. Available tasks = {1}. - At time = 3, task 2 is available to process. Available tasks = {1, 2}. - Also at time = 3, the CPU finishes task 0 and starts processing task 2 as it is the shortest. Available tasks = {1}. - At time = 4, task 3 is available to process. Available tasks = {1, 3}. - At time = 5, the CPU finishes task 2 and starts processing task 3 as it is the shortest. Available tasks = {1}. - At time = 6, the CPU finishes task 3 and starts processing task 1. Available tasks = {}. - At time = 10, the CPU finishes task 1 and becomes idle.

Example 2: Input: tasks = [[7,10],[7,12],[7,5],[7,4],[7,2]] Output: [4,3,2,0,1] Explanation: The events go as follows: - At time = 7, all the tasks become available. Available tasks = {0,1,2,3,4}. - Also at time = 7, the idle CPU starts processing task 4. Available tasks = {0,1,2,3}. - At time = 9, the CPU finishes task 4 and starts processing task 3. Available tasks = {0,1,2}. - At time = 13, the CPU finishes task 3 and starts processing task 2. Available tasks = {0,1}. - At time = 18, the CPU finishes task 2 and starts processing task 0. Available tasks = {1}. - At time = 28, the CPU finishes task 0 and starts processing task 1. Available tasks = {}. - At time = 40, the CPU finishes task 1 and becomes idle.

Constraints:

tasks.length == n 1 <= n <= 105 1 <= enqueueTimei, processingTimei <= 109

==
Problem Number: 1292 URL: https://leetcode.com/problems/frequency-of-the-most-frequent-element Title: 1838. Frequency of the Most Frequent Element Problem Description: The frequency of an element is the number of times it occurs in an array. You are given an integer array nums and an integer k. In one operation, you can choose an index of nums and increment the element at that index by 1. Return the maximum possible frequency of an element after performing at most k operations. Example 1: Input: nums = [1,2,4], k = 5 Output: 3 Explanation: Increment the first element three times and the second element two times to make nums = [4,4,4]. 4 has a frequency of 3. Example 2: Input: nums = [1,4,8,13], k = 5 Output: 2 Explanation: There are multiple optimal solutions: - Increment the first element three times to make nums = [4,4,8,13]. 4 has a frequency of 2. - Increment the second element four times to make nums = [1,8,8,13]. 8 has a frequency of 2. - Increment the third element five times to make nums = [1,4,13,13]. 13 has a frequency of 2.

Example 3: Input: nums = [3,9,6], k = 2 Output: 1

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105 1 <= k <= 105

==
Problem Number: 1293 URL: https://leetcode.com/problems/longest-substring-of-all-vowels-in-order Title: 1839. Longest Substring Of All Vowels in Order Problem Description: A string is considered beautiful if it satisfies the following conditions:

Each of the 5 English vowels ('a', 'e', 'i', 'o', 'u') must appear at least once in it. The letters must be sorted in alphabetical order (i.e. all 'a's before 'e's, all 'e's before 'i's, etc.).

For example, strings ”aeiou” and ”aaaaaaeiiiioou” are considered beautiful, but ”uaeio”, ”aeoiu”, and ”aaaeeeooo” are not beautiful. Given a string word consisting of English vowels, return the length of the longest beautiful substring of word. If no such substring exists, return 0. A substring is a contiguous sequence of characters in a string. Example 1: Input: word = ”aeiaaioaaaaeiiiiouuuooaauuaeiu” Output: 13 Explanation: The longest beautiful substring in word is ”aaaaeiiiiouuu” of length 13. Example 2: Input: word = ”aeeeiiiioooauuuaeiou” Output: 5 Explanation: The longest beautiful substring in word is ”aeiou” of length 5.

Example 3: Input: word = ”a” Output: 0 Explanation: There is no beautiful substring, so return 0.

Constraints:

1 <= word.length <= 5 * 105 word consists of characters ’a’, ’e’, ’i’, ’o’, and ’u’.

==

Problem Number: 1294 URL: https://leetcode.com/problems/seat-reservation-manager Title: 1845. Seat Reservation Manager Problem Description: Design a system that manages the reservation state of n seats that are numbered from 1 to n. Implement the SeatManager class:

SeatManager(int n) Initializes a SeatManager object that will manage n seats numbered from 1 to n. All seats are initially available. int reserve() Fetches the smallest-numbered unreserved seat, reserves it, and returns its number. void unreserve(int seatNumber) Unreserves the seat with the given seatNumber.

Example 1: Input [”SeatManager”, ”reserve”, ”reserve”, ”unreserve”, ”reserve”, ”reserve”, ”reserve”, ”reserve”, ”unreserve”] [[5], [], [], [2], [], [], [], [], [5]] Output [null, 1, 2, null, 2, 3, 4, 5, null]

Explanation SeatManager seatManager = new SeatManager(5); // Initializes a SeatManager with 5 seats. seatManager.reserve(); // All seats are available, so return the lowest numbered seat, which is 1. seatManager.reserve(); // The available seats are [2,3,4,5], so return the lowest of them, which is 2. seatManager.unreserve(2); // Unreserve seat 2, so now the available seats are [2,3,4,5]. seatManager.reserve(); // The available seats are [2,3,4,5], so return the lowest of them, which is 2. seatManager.reserve(); // The available seats are [3,4,5], so return the lowest of them, which is 3. seatManager.reserve(); // The available seats are [4,5], so return the lowest of them, which is 4. seatManager.reserve(); // The only available seat is seat 5, so return 5. seatManager.unreserve(5); // Unreserve seat 5, so now the available seats are [5].

Constraints:

1 <= n <= 105 1 <= seatNumber <= n For each call to reserve, it is guaranteed that there will be at least one unreserved seat. For each call to unreserve, it is guaranteed that seatNumber will be reserved. At most 105 calls in total will be made to reserve and unreserve.

==
Problem Number: 1295 URL: https://leetcode.com/problems/maximum-element-after-decreasing-and-rearranging Title: 1846. Maximum Element After Decreasing and Rearranging Problem Description: You are given an array of positive integers arr. Perform some operations (possibly none) on arr so that it satisfies these conditions:

The value of the first element in arr must be 1. The absolute difference between any 2 adjacent elements must be less than or equal to 1. In other words, abs(arr[i] - arr[i - 1]) <= 1 for each i where 1 <= i < arr.length (0-indexed). abs(x) is the absolute value of x.

There are 2 types of operations that you can perform any number of times:

Decrease the value of any element of arr to a smaller positive integer. Rearrange the elements of arr to be in any order.

Return the maximum possible value of an element in arr after performing the operations to satisfy the conditions. Example 1: Input: arr = [2,2,1,2,1] Output: 2 Explanation: We can satisfy the conditions by rearranging arr so it becomes [1,2,2,2,1]. The largest element in arr is 2.

Example 2: Input: arr = [100,1,1000] Output: 3 Explanation: One possible way to satisfy the conditions is by doing the following: 1. Rearrange arr so it becomes [1,100,1000]. 2. Decrease the value of the second element to 2. 3. Decrease the value of the third element to 3. Now arr = [1,2,3], which satisfies the conditions. The largest element in arr is 3.

Example 3: Input: arr = [1,2,3,4,5] Output: 5 Explanation: The array already satisfies the conditions, and the largest element is 5.

Constraints:

1 <= arr.length <= 105 1 <= arr[i] <= 109

==
Problem Number: 1296 URL: https://leetcode.com/problems/splitting-a-string-into-descending-consecutive-values Title: 1849. Splitting a String Into Descending Consecutive Values Problem Description: You are given a string s that consists of only digits. Check if we can split s into two or more non-empty substrings such that the numerical values of the substrings are in descending order and the difference between numerical values of every two adjacent substrings is equal to 1.

For example, the string s = "0090089" can be split into ["0090", "089"] with numerical values [90,89]. The values are in descending order and adjacent values differ by 1, so this way is valid. Another example, the string s = "001" can be split into ["0", "01"], ["00", "1"], or ["0", "0", "1"]. However all the ways are invalid because they have numerical values [0,1], [0,1], and [0,0,1] respectively, all of which are not in descending order.

Return true if it is possible to split s as described above, or false otherwise. A substring is a contiguous sequence of characters in a string.   Example 1: Input: s = ”1234” Output: false Explanation: There is no valid way to split s.

Example 2: Input: s = ”050043” Output: true Explanation: s can be split into [”05”, ”004”, ”3”] with numerical values [5,4,3]. The values are in descending order with adjacent values differing by 1.

Example 3: Input: s = ”9080701” Output: false Explanation: There is no valid way to split s.

Constraints:

1 <= s.length <= 20 s only consists of digits.

==
Problem Number: 1297 URL: https://leetcode.com/problems/minimum-adjacent-swaps-to-reach-the-kth-smallest-number Title: 1850. Minimum Adjacent Swaps to Reach the Kth Smallest Number Problem Description: You are given a string num, representing a large integer, and an integer k. We call some integer wonderful if it is a permutation of the digits in num and is greater in value than num. There can be many wonderful integers. However, we only care about the smallest-valued ones.

For example, when num = ”5489355142”:

The 1st smallest wonderful integer is ”5489355214”. The 2nd smallest wonderful integer is ”5489355241”. The 3rd smallest wonderful integer is ”5489355412”. The 4th smallest wonderful integer is ”5489355421”.

Return the minimum number of adjacent digit swaps that needs to be applied to num to reach the kth smallest wonderful integer. The tests are generated in such a way that kth smallest wonderful integer exists.   Example 1: Input: num = ”5489355142”, k = 4 Output: 2 Explanation: The 4th smallest wonderful number is ”5489355421”. To get this number: - Swap index 7 with index 8: ”5489355142” -> ”5489355412” - Swap index 8 with index 9: ”5489355412” -> ”5489355421”

Example 2: Input: num = ”11112”, k = 4 Output: 4 Explanation: The 4th smallest wonderful number is ”21111”. To get this number: - Swap index 3 with index 4: ”11112” -> ”11121” - Swap index 2 with index 3: ”11121” -> ”11211” - Swap index 1 with index 2: ”11211” -> ”12111” - Swap index 0 with index 1: ”12111” -> ”21111”

Example 3: Input: num = ”00123”, k = 1 Output: 1 Explanation: The 1st smallest wonderful number is ”00132”. To get this number: - Swap index 3 with index 4: ”00123” -> ”00132”

Constraints:

2 <= num.length <= 1000 1 <= k <= 1000 num only consists of digits.

==
Problem Number: 1298 URL: https://leetcode.com/problems/maximum-distance-between-a-pair-of-values Title: 1855. Maximum Distance Between a Pair of Values Problem Description: You are given two non-increasing 0-indexed integer arrays nums1 and nums2. A pair of indices (i, j), where 0 <= i < nums1.length and 0 <= j < nums2.length, is valid if both i <= j and nums1[i] <= nums2[j]. The distance of the pair is j - i. Return the maximum distance of any valid pair (i, j). If there are no valid pairs, return 0. An array arr is non-increasing if arr[i-1] >= arr[i] for every 1 <= i < arr.length. Example 1: Input: nums1 = [55,30,5,4,2], nums2 = [100,20,10,10,5] Output: 2 Explanation: The valid pairs are (0,0), (2,2), (2,3), (2,4), (3,3), (3,4), and (4,4). The maximum distance is 2 with pair (2,4).

Example 2: Input: nums1 = [2,2,2], nums2 = [10,10,1] Output: 1 Explanation: The valid pairs are (0,0), (0,1), and (1,1). The maximum distance is 1 with pair (0,1).

Example 3: Input: nums1 = [30,29,19,5], nums2 = [25,25,25,25,25] Output: 2 Explanation: The valid pairs are (2,2), (2,3), (2,4), (3,3), and (3,4). The maximum distance is 2 with pair (2,4).

Constraints:

1 <= nums1.length, nums2.length <= 105 1 <= nums1[i], nums2[j] <= 105 Both nums1 and nums2 are non-increasing.

==
Problem Number: 1299 URL: https://leetcode.com/problems/maximum-subarray-min-product Title: 1856. Maximum Subarray Min-Product Problem Description: The min-product of an array is equal to the minimum value in the array multiplied by the array's sum.

For example, the array [3,2,5] (minimum value is 2) has a min-product of 2 * (3+2+5) = 2 * 10 = 20.

Given an array of integers nums, return the maximum min-product of any non-empty subarray of nums. Since the answer may be large, return it modulo 109 + 7. Note that the min-product should be maximized before performing the modulo operation. Testcases are generated such that the maximum min-product without modulo will fit in a 64-bit signed integer. A subarray is a contiguous part of an array. Example 1: Input: nums = [1,2,3,2] Output: 14 Explanation: The maximum min-product is achieved with the subarray [2,3,2] (minimum value is 2). 2 * (2+3+2) = 2 * 7 = 14.

Example 2: Input: nums = [2,3,3,1,2] Output: 18 Explanation: The maximum min-product is achieved with the subarray [3,3] (minimum value is 3). 3 * (3+3) = 3 * 6 = 18.

Example 3: Input: nums = [3,1,5,6,4,2] Output: 60 Explanation: The maximum min-product is achieved with the subarray [5,6,4] (minimum value is 4). 4 *

(5+6+4) = 4 * 15 = 60.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 107

== Problem Number: 1300 URL: https://leetcode.com/problems/incremental-memory-leak Title: 1860. Incremental Memory Leak Problem Description: You are given two integers memory1 and memory2 representing the available memory in bits on two memory sticks. There is currently a faulty program running that consumes an increasing amount of memory every second. At the ith second (starting from 1), i bits of memory are allocated to the stick with more available memory (or from the first memory stick if both have the same available memory). If neither stick has at least i bits of available memory, the program crashes. Return an array containing [crashTime, memory1crash, memory2crash], where crashTime is the time (in seconds) when the program crashed and memory1crash and memory2crash are the available bits of memory in the first and second sticks respectively. Example 1: Input: memory1 = 2, memory2 = 2 Output: [3,1,0] Explanation: The memory is allocated as follows: - At the 1st second, 1 bit of memory is allocated to stick 1. The first stick now has 1 bit of available memory. - At the 2nd second, 2 bits of memory are allocated to stick 2. The second stick now has 0 bits of available memory. - At the 3rd second, the program crashes. The sticks have 1 and 0 bits available respectively.

Example 2: Input: memory1 = 8, memory2 = 11 Output: [6,0,4] Explanation: The memory is allocated as follows: - At the 1st second, 1 bit of memory is allocated to stick 2. The second stick now has 10 bit of available memory. - At the 2nd second, 2 bits of memory are allocated to stick 2. The second stick now has 8 bits of available memory. - At the 3rd second, 3 bits of memory are allocated to stick 1. The first stick now has 5 bits of available memory. - At the 4th second, 4 bits of memory are allocated to stick 2. The second stick now has 4 bits of available memory. - At the 5th second, 5 bits of memory are allocated to stick 1. The first stick now has 0 bits of available memory. - At the 6th second, the program crashes. The sticks have 0 and 4 bits available respectively.

Constraints:

0 <= memory1, memory2 <= 231 - 1

== Problem Number: 1301 URL: https://leetcode.com/problems/rotating-the-box Title: 1861. Rotating the Box Problem Description: You are given an m x n matrix of characters box representing a side-view of a box. Each cell of the box is one of the following:

A stone '#' A stationary obstacle '*' Empty '.'

The box is rotated 90 degrees clockwise, causing some of the stones to fall due to gravity. Each stone falls down until it lands on an obstacle, another stone, or the bottom of the box. Gravity does not affect the obstacles' positions, and the inertia from the box's rotation does not affect the stones' horizontal positions. It is guaranteed that each stone in box rests on an obstacle, another stone, or the bottom of the box. Return an n x m matrix representing the box after the rotation described above. Example 1:

Input: box = [["#",".","#"]] Output: [["."], ["#"], ["#"]]

Example 2:

Input: box = [["#",".","*","."], ["#","#","*","."]] Output: [["#","."], ["#","#"], ["*","*"], [".","."]]

Example 3:

Input: box = [["#","#","*",".","*","."], ["#","#","#","*",".","."], ["#","#","#",".","#","."]] Output: [[".","#","#"], [".","#","#"], ["#","#","*"], ["#","*","."], ["#",".","*"], ["#",".","."]]

Constraints:

m == box.length n == box[i].length 1 <= m, n <= 500 box[i][j] is either '#', '*', or '.'.

==
Problem Number: 1302 URL: https://leetcode.com/problems/minimum-number-of-swaps-to-make-the-binary-string-alternating Title: 1864. Minimum Number of Swaps to Make the Binary String Alternating Problem Description: Given a binary string s, return the minimum number of character swaps to make it alternating, or -1 if it is impossible. The string is called alternating if no two adjacent characters are equal. For example, the strings "010" and "1010" are alternating, while the string "0100" is not. Any two characters may be swapped, even if they are not adjacent. Example 1: Input: s = "111000" Output: 1 Explanation: Swap positions 1 and 4: "111000" -> "101010" The string is now alternating.

Example 2: Input: s = "010" Output: 0 Explanation: The string is already alternating, no swaps are needed.

Example 3: Input: s = "1110" Output: -1

Constraints:

1 <= s.length <= 1000 s[i] is either '0' or '1'.

==
Problem Number: 1303 URL: https://leetcode.com/problems/finding-pairs-with-a-certain-sum Title: 1865. Finding Pairs With a Certain Sum Problem Description: You are given two integer arrays nums1 and nums2. You are tasked to implement a data structure that supports queries of two types:

Add a positive integer to an element of a given index in the array nums2. Count the number of pairs (i, j) such that nums1[i] + nums2[j] equals a given value (0 <= i < nums1.length and 0 <= j < nums2.length).

Implement the FindSumPairs class:

FindSumPairs(int[] nums1, int[] nums2) Initializes the FindSumPairs object with two integer arrays nums1 and nums2. void add(int index, int val) Adds val to nums2[index], i.e., apply nums2[index] += val. int count(int tot) Returns the number of pairs (i, j) such that nums1[i] + nums2[j] == tot.

Example 1: Input ["FindSumPairs", "count", "add", "count", "count", "add", "add", "count"] [[[1, 1, 2, 2, 2, 3], [1, 4, 5, 2, 5, 4]], [7], [3, 2], [8], [4], [0, 1], [1, 1], [7]] Output [null, 8, null, 2, 1, null, null, 11]

Explanation FindSumPairs findSumPairs = new FindSumPairs([1, 1, 2, 2, 2, 3], [1, 4, 5, 2, 5, 4]); findSumPairs.count(7); // return 8; pairs (2,2), (3,2), (4,2), (2,4), (3,4), (4,4) make 2 + 5 and pairs (5,1), (5,5) make 3 + 4 findSumPairs.add(3, 2); // now nums2 = [1,4,5,4,5,4] findSumPairs.count(8); // return 2; pairs (5,2), (5,4) make 3 + 5 findSumPairs.count(4); // return 1; pair (5,0) makes 3 + 1 findSumPairs.add(0, 1); // now nums2 = [2,4,5,4,5,4] findSumPairs.add(1, 1); // now nums2 = [2,5,5,4,5,4] findSumPairs.count(7); // return 11; pairs (2,1), (2,2), (2,4), (3,1), (3,2), (3,4), (4,1), (4,2), (4,4) make 2 + 5 and pairs (5,3), (5,5) make 3 + 4

Constraints:

1 <= nums1.length <= 1000 1 <= nums2.length <= 105 1 <= nums1[i] <= 109 1 <= nums2[i] <= 105 0 <= index < nums2.length 1 <= val <= 105 1 <= tot <= 109 At most 1000 calls are made to add and count each.

==

Problem Number: 1304 URL: https://leetcode.com/problems/minimum-speed-to-arrive-on-time Title: 1870. Minimum Speed to Arrive on Time Problem Description: You are given a floating-point number hour, representing the amount of time you have to reach the office. To commute to the office, you must take n trains in sequential order. You are also given an integer array dist of length n, where dist[i] describes the distance (in kilometers) of the ith train ride. Each train can only depart at an integer hour, so you may need to wait in between each train ride.

For example, if the 1st train ride takes 1.5 hours, you must wait for an additional 0.5 hours before you can depart on the 2nd train ride at the 2 hour mark.

Return the minimum positive integer speed (in kilometers per hour) that all the trains must travel at for you to reach the office on time, or -1 if it is impossible to be on time. Tests are generated such that the answer will not exceed 107 and hour will have at most two digits after the decimal point. Example 1: Input: dist = [1,3,2], hour = 6 Output: 1 Explanation: At speed 1: - The first train ride takes 1/1 = 1 hour. - Since we are already at an integer hour, we depart immediately at the 1 hour mark. The second train takes 3/1 = 3 hours. - Since we are already at an integer hour, we depart immediately at the 4 hour mark. The third train takes 2/1 = 2 hours. - You will arrive at exactly the 6 hour mark.

Example 2: Input: dist = [1,3,2], hour = 2.7 Output: 3 Explanation: At speed 3: - The first train ride takes 1/3 = 0.33333 hours. - Since we are not at an integer hour, we wait until the 1 hour mark to depart. The second train ride takes 3/3 = 1 hour. - Since we are already at an integer hour, we depart immediately at the 2 hour mark. The third train takes 2/3 = 0.66667 hours. - You will arrive at the 2.66667 hour mark.

Example 3: Input: dist = [1,3,2], hour = 1.9 Output: -1 Explanation: It is impossible because the earliest the third train can depart is at the 2 hour mark.

Constraints:

n == dist.length 1 <= n <= 105 1 <= dist[i] <= 105 1 <= hour <= 109 There will be at most two digits after the decimal point in hour.

==
Problem Number: 1305 URL: https://leetcode.com/problems/jump-game-vii Title: 1871. Jump Game VII Problem Description: You are given a 0-indexed binary string s and two integers minJump and maxJump. In the beginning, you are standing at index 0, which is equal to '0'. You can move from index i to index j if the following conditions are fulfilled:

i + minJump <= j <= min(i + maxJump, s.length - 1), and s[j] == '0'.

Return true if you can reach index s.length - 1 in s, or false otherwise. Example 1: Input: s = "011010", minJump = 2, maxJump = 3 Output: true Explanation: In the first step, move from index 0 to index 3. In the second step, move from index 3 to index 5.

Example 2: Input: s = "01101110", minJump = 2, maxJump = 3 Output: false

Constraints:

2 <= s.length <= 105 s[i] is either '0' or '1'. s[0] == '0' 1 <= minJump <= maxJump < s.length

==
Problem Number: 1306 URL: https://leetcode.com/problems/minimize-maximum-pair-sum-in-array Title: 1877. Minimize Maximum Pair Sum in Array Problem Description: The pair sum of a pair (a,b) is equal to a + b. The maximum pair sum is the largest pair sum in a list of pairs.

For example, if we have pairs (1,5), (2,3), and (4,4), the maximum pair sum would be max(1+5, 2+3, 4+4) = max(6, 5, 8) = 8.

Given an array nums of even length n, pair up the elements of nums into n / 2 pairs such that:

Each element of nums is in exactly one pair, and The maximum pair sum is minimized.

Return the minimized maximum pair sum after optimally pairing up the elements. Example 1: Input: nums = [3,5,2,3] Output: 7 Explanation: The elements can be paired up into pairs (3,3) and (5,2). The maximum pair sum is max(3+3, 5+2) = max(6, 7) = 7.

Example 2: Input: nums = [3,5,4,2,4,6] Output: 8 Explanation: The elements can be paired up into pairs (3,5), (4,4), and (6,2). The maximum pair sum is max(3+5, 4+4, 6+2) = max(8, 8, 8) = 8.

Constraints:

n == nums.length 2 <= n <= 105 n is even. 1 <= nums[i] <= 105

==
Problem Number: 1307 URL: https://leetcode.com/problems/get-biggest-three-rhombus-sums-in-a-grid Title: 1878. Get Biggest Three Rhombus Sums in a Grid Problem Description: You are given an m x n integer matrix grid. A rhombus sum is the sum of the elements that form the border of a regular rhombus shape in grid. The rhombus must have the shape of a square rotated 45 degrees with each of the corners centered in a grid cell. Below is an image of four valid rhombus shapes with the corresponding colored cells that should be included in each rhombus sum:

Note that the rhombus can have an area of 0, which is depicted by the purple rhombus in the bottom right corner. Return the biggest three distinct rhombus sums in the grid in descending order. If there are less than three distinct values, return all of them. Example 1:

Input: grid = [[3,4,5,1,3],[3,3,4,2,3],[20,30,200,40,10],[1,5,5,4,1],[4,3,2,2,5]] Output: [228,216,211] Explanation: The rhombus shapes for the three biggest distinct rhombus sums are depicted above. - Blue: 20 + 3 + 200 + 5 = 228 - Red: 200 + 2 + 10 + 4 = 216 - Green: 5 + 200 + 4 + 2 = 211

Example 2:

Input: grid = [[1,2,3],[4,5,6],[7,8,9]] Output: [20,9,8] Explanation: The rhombus shapes for the three biggest distinct rhombus sums are depicted above. - Blue: 4 + 2 + 6 + 8 = 20 - Red: 9 (area 0 rhombus in the bottom right corner) - Green: 8 (area 0 rhombus in the bottom middle)

Example 3: Input: grid = [[7,7,7]] Output: [7] Explanation: All three possible rhombus sums are the same, so return [7].

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 50 1 <= grid[i][j] <= 105

==
Problem Number: 1308 URL: https://leetcode.com/problems/maximum-

value-after-insertion Title: 1881. Maximum Value after Insertion Problem Description: You are given a very large integer n, represented as a string, and an integer digit x. The digits in n and the digit x are in the inclusive range [1, 9], and n may represent a negative number. You want to maximize n's numerical value by inserting x anywhere in the decimal representation of n. You cannot insert x to the left of the negative sign.

For example, if n = 73 and x = 6, it would be best to insert it between 7 and 3, making n = 763. If n = -55 and x = 2, it would be best to insert it before the first 5, making n = -255.

Return a string representing the maximum value of n after the insertion. Example 1: Input: n = "99", x = 9 Output: "999" Explanation: The result is the same regardless of where you insert 9.

Example 2: Input: n = "-13", x = 2 Output: "-123" Explanation: You can make n one of {-213, -123, -132}, and the largest of those three is -123.

Constraints:

1 <= n.length <= 105 1 <= x <= 9 The digits in n are in the range [1, 9]. n is a valid representation of an integer. In the case of a negative n, it will begin with '-'.

==
Problem Number: 1309 URL: https://leetcode.com/problems/process-tasks-using-servers Title: 1882. Process Tasks Using Servers Problem Description: You are given two 0-indexed integer arrays servers and tasks of lengths n and m respectively. servers[i] is the weight of the ith server, and tasks[j] is the time needed to process the jth task in seconds. Tasks are assigned to the servers using a task queue. Initially, all servers are free, and the queue is empty. At second j, the jth task is inserted into the queue (starting with the 0th task being inserted at second 0). As long as there are free servers and the queue is not empty, the task in the front of the queue will be assigned to a free server with the smallest weight, and in case of a tie, it is assigned to a free server with the smallest index. If there are no free servers and the queue is not empty, we wait until a server becomes free and immediately assign the next task. If multiple servers become free at the same time, then multiple tasks from the queue will be assigned in order of insertion following the weight and index priorities above. A server that is assigned task j at second t will be free again at second t + tasks[j]. Build an array ans of length m, where ans[j] is the index of the server the jth task will be assigned to. Return the array ans. Example 1: Input: servers = [3,3,2], tasks = [1,2,3,2,1,2] Output: [2,2,0,2,1,2] Explanation: Events in chronological order go as follows: - At second 0, task 0 is added and processed using server 2 until second 1. - At second 1, server 2 becomes free. Task 1 is added and processed using server 2 until second 3. - At second 2, task 2 is added and processed using server 0 until second 5. - At second 3, server 2 becomes free. Task 3 is added and processed using server 2 until second 5. - At second 4, task 4 is added and processed using server 1 until

second 5. - At second 5, all servers become free. Task 5 is added and processed using server 2 until second 7. Example 2: Input: servers = [5,1,4,3,2], tasks = [2,1,2,4,5,2,1] Output: [1,4,1,4,1,3,2] Explanation: Events in chronological order go as follows: - At second 0, task 0 is added and processed using server 1 until second 2. - At second 1, task 1 is added and processed using server 4 until second 2. - At second 2, servers 1 and 4 become free. Task 2 is added and processed using server 1 until second 4. - At second 3, task 3 is added and processed using server 4 until second 7. - At second 4, server 1 becomes free. Task 4 is added and processed using server 1 until second 9. - At second 5, task 5 is added and processed using server 3 until second 7. - At second 6, task 6 is added and processed using server 2 until second 7.

Constraints:

servers.length == n tasks.length == m 1 <= n, m <= 2 * 105 1 <= servers[i], tasks[j] <= 2 * 105

==

Problem Number: 1310 URL: https://leetcode.com/problems/egg-drop-with-2-eggs-and-n-floors Title: 1884. Egg Drop With 2 Eggs and N Floors Problem Description: You are given two identical eggs and you have access to a building with n floors labeled from 1 to n. You know that there exists a floor f where 0 <= f <= n such that any egg dropped at a floor higher than f will break, and any egg dropped at or below floor f will not break. In each move, you may take an unbroken egg and drop it from any floor x (where 1 <= x <= n). If the egg breaks, you can no longer use it. However, if the egg does not break, you may reuse it in future moves. Return the minimum number of moves that you need to determine with certainty what the value of f is. Example 1: Input: n = 2 Output: 2 Explanation: We can drop the first egg from floor 1 and the second egg from floor 2. If the first egg breaks, we know that f = 0. If the second egg breaks but the first egg didn't, we know that f = 1. Otherwise, if both eggs survive, we know that f = 2.

Example 2: Input: n = 100 Output: 14 Explanation: One optimal strategy is: - Drop the 1st egg at floor 9. If it breaks, we know f is between 0 and 8. Drop the 2nd egg starting from floor 1 and going up one at a time to find f within 8 more drops. Total drops is 1 + 8 = 9. - If the 1st egg does not break, drop the 1st egg again at floor 22. If it breaks, we know f is between 9 and 21. Drop the 2nd egg starting from floor 10 and going up one at a time to find f within 12 more drops. Total drops is 2 + 12 = 14. - If the 1st egg does not break again, follow a similar process dropping the 1st egg from floors 34, 45, 55, 64, 72, 79, 85, 90, 94, 97, 99, and 100. Regardless of the outcome, it takes at most 14 drops to determine f.

Constraints:

1 <= n <= 1000

==

Problem Number: 1311 URL: https://leetcode.com/problems/reduction-operations-to-make-the-array-elements-equal Title: 1887. Reduction Operations to Make the Array Elements Equal Problem Description: Given an integer array nums, your goal is to make all elements in nums equal. To complete one operation, follow these steps:

Find the largest value in nums. Let its index be i (0-indexed) and its value be largest. If there are multiple elements with the largest value, pick the smallest i. Find the next largest value in nums strictly smaller than largest. Let its value be nextLargest. Reduce nums[i] to nextLargest.

Return the number of operations to make all elements in nums equal. Example 1: Input: nums = [5,1,3] Output: 3 Explanation: It takes 3 operations to make all elements in nums equal: 1. largest = 5 at index 0. nextLargest = 3. Reduce nums[0] to 3. nums = [3,1,3]. 2. largest = 3 at index 0. nextLargest = 1. Reduce nums[0] to 1. nums = [1,1,3]. 3. largest = 3 at index 2. nextLargest = 1. Reduce nums[2] to 1. nums = [1,1,1].

Example 2: Input: nums = [1,1,1] Output: 0 Explanation: All elements in nums are already equal.

Example 3: Input: nums = [1,1,2,2,3] Output: 4 Explanation: It takes 4 operations to make all elements in nums equal: 1. largest = 3 at index 4. nextLargest = 2. Reduce nums[4] to 2. nums = [1,1,2,2,2]. 2. largest = 2 at index 2. nextLargest = 1. Reduce nums[2] to 1. nums = [1,1,1,2,2]. 3. largest = 2 at index 3. nextLargest = 1. Reduce nums[3] to 1. nums = [1,1,1,1,2]. 4. largest = 2 at index 4. nextLargest = 1. Reduce nums[4] to 1. nums = [1,1,1,1,1].

Constraints:

1 <= nums.length <= 5 * 104 1 <= nums[i] <= 5 * 104

==

Problem Number: 1312 URL: https://leetcode.com/problems/minimum-number-of-flips-to-make-the-binary-string-alternating Title: 1888. Minimum Number of Flips to Make the Binary String Alternating Problem Description: You are given a binary string s. You are allowed to perform two types of operations on the string in any sequence:

Type-1: Remove the character at the start of the string s and append it to the end of the string. Type-2: Pick any character in s and flip its value, i.e., if its value is '0' it becomes '1' and vice-versa.

Return the minimum number of type-2 operations you need to perform such that s becomes alternating. The string is called alternating if no two adjacent characters are equal.

For example, the strings "010" and "1010" are alternating, while the string "0100" is not.

Example 1: Input: s = ”111000” Output: 2 Explanation: Use the first operation two times to make s = ”100011”. Then, use the second operation on the third and sixth elements to make s = ”101010”.

Example 2: Input: s = ”010” Output: 0 Explanation: The string is already alternating.

Example 3: Input: s = ”1110” Output: 1 Explanation: Use the second operation on the second element to make s = ”1010”.

Constraints:

1 <= s.length <= 105 s[i] is either ’0’ or ’1’.

==

Problem Number: 1313 URL: https://leetcode.com/problems/find-the-student-that-will-replace-the-chalk Title: 1894. Find the Student that Will Replace the Chalk Problem Description: There are n students in a class numbered from 0 to n - 1. The teacher will give each student a problem starting with the student number 0, then the student number 1, and so on until the teacher reaches the student number n - 1. After that, the teacher will restart the process, starting with the student number 0 again. You are given a 0-indexed integer array chalk and an integer k. There are initially k pieces of chalk. When the student number i is given a problem to solve, they will use chalk[i] pieces of chalk to solve that problem. However, if the current number of chalk pieces is strictly less than chalk[i], then the student number i will be asked to replace the chalk. Return the index of the student that will replace the chalk pieces. Example 1: Input: chalk = [5,1,5], k = 22 Output: 0 Explanation: The students go in turns as follows: - Student number 0 uses 5 chalk, so k = 17. - Student number 1 uses 1 chalk, so k = 16. - Student number 2 uses 5 chalk, so k = 11. - Student number 0 uses 5 chalk, so k = 6. - Student number 1 uses 1 chalk, so k = 5. - Student number 2 uses 5 chalk, so k = 0. Student number 0 does not have enough chalk, so they will have to replace it. Example 2: Input: chalk = [3,4,1,2], k = 25 Output: 1 Explanation: The students go in turns as follows: - Student number 0 uses 3 chalk so k = 22. - Student number 1 uses 4 chalk so k = 18. - Student number 2 uses 1 chalk so k = 17. - Student number 3 uses 2 chalk so k = 15. - Student number 0 uses 3 chalk so k = 12. - Student number 1 uses 4 chalk so k = 8. - Student number 2 uses 1 chalk so k = 7. - Student number 3 uses 2 chalk so k = 5. - Student number 0 uses 3 chalk so k = 2. Student number 1 does not have enough chalk, so they will have to replace it.

Constraints:

chalk.length == n 1 <= n <= 105 1 <= chalk[i] <= 105 1 <= k <= 109

==

Problem Number: 1314 URL: https://leetcode.com/problems/largest-magic-square Title: 1895. Largest Magic Square Problem Description: A k x k magic square is a k x k grid filled with integers such that every row sum, every column sum, and both diagonal sums are all equal. The integers in the magic square do not have to be distinct. Every 1 x 1 grid is trivially a magic square. Given an m x n integer grid, return the size (i.e., the side length k) of the largest magic square that can be found within this grid. Example 1:

Input: grid = [[7,1,4,5,6],[2,5,1,6,4],[1,5,4,3,2],[1,2,7,3,4]] Output: 3 Explanation: The largest magic square has a size of 3. Every row sum, column sum, and diagonal sum of this magic square is equal to 12. - Row sums: 5+1+6 = 5+4+3 = 2+7+3 = 12 - Column sums: 5+5+2 = 1+4+7 = 6+3+3 = 12 - Diagonal sums: 5+4+3 = 6+4+2 = 12

Example 2:

Input: grid = [[5,1,3,1],[9,3,3,1],[1,3,3,8]] Output: 2

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 50 1 <= grid[i][j] <= 106

==
Problem Number: 1315 URL: https://leetcode.com/problems/maximum-number-of-removable-characters Title: 1898. Maximum Number of Removable Characters Problem Description: You are given two strings s and p where p is a subsequence of s. You are also given a distinct 0-indexed integer array removable containing a subset of indices of s (s is also 0-indexed). You want to choose an integer k (0 <= k <= removable.length) such that, after removing k characters from s using the first k indices in removable, p is still a subsequence of s. More formally, you will mark the character at s[removable[i]] for each 0 <= i < k, then remove all marked characters and check if p is still a subsequence. Return the maximum k you can choose such that p is still a subsequence of s after the removals. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters. Example 1: Input: s = "abcacb", p = "ab", removable = [3,1,0] Output: 2 Explanation: After removing the characters at indices 3 and 1, "abcacb" becomes "accb". "ab" is a subsequence of "accb". If we remove the characters at indices 3, 1, and 0, "abcacb" becomes "ccb", and "ab" is no longer a subsequence. Hence, the maximum k is 2.

Example 2: Input: s = "abcbddddd", p = "abcd", removable = [3,2,1,4,5,6] Output: 1 Explanation: After removing the character at index 3, "abcbddddd" becomes "abcddddd". "abcd" is a subsequence of "abcddddd".

Example 3: Input: s = "abcab", p = "abc", removable = [0,1,2,3,4] Output: 0 Explanation: If you remove the first index in the array removable, "abc" is no longer a subsequence.

Constraints:

1 <= p.length <= s.length <= 105 0 <= removable.length < s.length 0 <= removable[i] < s.length p is a subsequence of s. s and p both consist of lowercase English letters. The elements in removable are distinct.

===
Problem Number: 1316 URL: https://leetcode.com/problems/merge-triplets-to-form-target-triplet Title: 1899. Merge Triplets to Form Target Triplet Problem Description: A triplet is an array of three integers. You are given a 2D integer array triplets, where triplets[i] = [ai, bi, ci] describes the ith triplet. You are also given an integer array target = [x, y, z] that describes the triplet you want to obtain. To obtain target, you may apply the following operation on triplets any number of times (possibly zero):

Choose two indices (0-indexed) i and j (i != j) and update triplets[j] to become [max(ai, aj), max(bi, bj), max(ci, cj)].

For example, if triplets[i] = [2, 5, 3] and triplets[j] = [1, 7, 5], triplets[j] will be updated to [max(2, 1), max(5, 7), max(3, 5)] = [2, 7, 5].

Return true if it is possible to obtain the target triplet [x, y, z] as an element of triplets, or false otherwise. Example 1: Input: triplets = [[2,5,3],[1,8,4],[1,7,5]], target = [2,7,5] Output: true Explanation: Perform the following operations: - Choose the first and last triplets [[2,5,3],[1,8,4],[1,7,5]]. Update the last triplet to be [max(2,1), max(5,7), max(3,5)] = [2,7,5]. triplets = [[2,5,3],[1,8,4],[2,7,5]] The target triplet [2,7,5] is now an element of triplets.

Example 2: Input: triplets = [[3,4,5],[4,5,6]], target = [3,2,5] Output: false Explanation: It is impossible to have [3,2,5] as an element because there is no 2 in any of the triplets.

Example 3: Input: triplets = [[2,5,3],[2,3,4],[1,2,5],[5,2,3]], target = [5,5,5] Output: true Explanation: Perform the following operations: - Choose the first and third triplets [[2,5,3],[2,3,4],[1,2,5],[5,2,3]]. Update the third triplet to be [max(2,1), max(5,2), max(3,5)] = [2,5,5]. triplets = [[2,5,3],[2,3,4],[2,5,5],[5,2,3]]. - Choose the third and fourth triplets [[2,5,3],[2,3,4],[2,5,5],[5,2,3]]. Update the fourth triplet to be [max(2,5), max(5,2), max(5,3)] = [5,5,5]. triplets = [[2,5,3],[2,3,4],[2,5,5],[5,5,5]]. The target triplet [5,5,5] is now an element of triplets.

Constraints:

1 <= triplets.length <= 105 triplets[i].length == target.length == 3 1 <= ai, bi, ci, x, y, z <= 1000

===
Problem Number: 1317 URL: https://leetcode.com/problems/find-a-peak-element-ii Title: 1901. Find a Peak Element II Problem Description: A peak element in a 2D grid is an element that is strictly greater than all of its adjacent neighbors to the left, right, top, and bottom. Given a 0-indexed m x n matrix mat where no two adjacent cells are equal, find any peak element mat[i][j] and return the length 2 array [i,j]. You may assume that the entire matrix is surrounded by an outer perimeter with the value -1 in each cell. You must write an algorithm that runs in O(m log(n)) or O(n log(m)) time. Example 1:

Input: mat = [[1,4],[3,2]] Output: [0,1] Explanation: Both 3 and 4 are peak elements so [1,0] and [0,1] are both acceptable answers.

Example 2:

Input: mat = [[10,20,15],[21,30,14],[7,16,32]] Output: [1,1] Explanation: Both 30 and 32 are peak elements so [1,1] and [2,2] are both acceptable answers.

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 500 1 <= mat[i][j] <= 105 No two adjacent cells are equal.

==
Problem Number: 1318 URL: https://leetcode.com/problems/the-number-of-full-rounds-you-have-played Title: 1904. The Number of Full Rounds You Have Played Problem Description: You are participating in an online chess tournament. There is a chess round that starts every 15 minutes. The first round of the day starts at 00:00, and after every 15 minutes, a new round starts.

For example, the second round starts at 00:15, the fourth round starts at 00:45, and the seventh round starts at 01:30.

You are given two strings loginTime and logoutTime where:

loginTime is the time you will login to the game, and logoutTime is the time you will logout from the game.

If logoutTime is earlier than loginTime, this means you have played from loginTime to midnight and from midnight to logoutTime. Return the number of full chess rounds you have played in the tournament. Note: All the given times follow the 24-hour clock. That means the first round of the day starts at 00:00 and the last round of the day starts at 23:45. Example 1: Input: loginTime = "09:31", logoutTime = "10:14" Output: 1 Explanation: You played one full round from 09:45 to 10:00. You did not play the full round from 09:30 to 09:45 because you logged in at 09:31 after it began. You did not play the full round from 10:00 to 10:15 because you logged out at 10:14 before it ended.

Example 2: Input: loginTime = "21:30", logoutTime = "03:00" Output: 22 Explanation: You played 10 full rounds from 21:30 to 00:00 and 12 full rounds from 00:00 to 03:00. 10 + 12 = 22.

Constraints:

loginTime and logoutTime are in the format hh:mm. 00 <= hh <= 23 00 <= mm <= 59 loginTime and logoutTime are not equal.

==
Problem Number: 1319 URL: https://leetcode.com/problems/count-sub-islands Title: 1905. Count Sub Islands Problem Description: You are given two m x n binary matrices grid1 and grid2 containing only 0's (representing water)

and 1's (representing land). An island is a group of 1's connected 4-directionally (horizontal or vertical). Any cells outside of the grid are considered water cells. An island in grid2 is considered a sub-island if there is an island in grid1 that contains all the cells that make up this island in grid2. Return the number of islands in grid2 that are considered sub-islands.   Example 1:

Input: grid1 = [[1,1,1,0,0],[0,1,1,1,1],[0,0,0,0,0],[1,0,0,0,0],[1,1,0,1,1]], grid2 = [[1,1,1,0,0],[0,0,1,1,1],[0,1,0,0,0],[1,0,1,1,0],[0,1,0,1,0]] Output: 3 Explanation: In the picture above, the grid on the left is grid1 and the grid on the right is grid2. The 1s colored red in grid2 are those considered to be part of a sub-island. There are three sub-islands.

Example 2:

Input: grid1 = [[1,0,1,0,1],[1,1,1,1,1],[0,0,0,0,0],[1,1,1,1,1],[1,0,1,0,1]], grid2 = [[0,0,0,0,0],[1,1,1,1,1],[0,1,0,1,0],[0,1,0,1,0],[1,0,0,0,1]] Output: 2 Explanation: In the picture above, the grid on the left is grid1 and the grid on the right is grid2. The 1s colored red in grid2 are those considered to be part of a sub-island. There are two sub-islands.

Constraints:

m == grid1.length == grid2.length n == grid1[i].length == grid2[i].length 1 <= m, n <= 500 grid1[i][j] and grid2[i][j] are either 0 or 1.

==
Problem Number: 1320 URL: https://leetcode.com/problems/minimum-absolute-difference-queries Title: 1906. Minimum Absolute Difference Queries Problem Description: The minimum absolute difference of an array a is defined as the minimum value of |a[i] - a[j]|, where 0 <= i < j < a.length and a[i] != a[j]. If all elements of a are the same, the minimum absolute difference is -1.

For example, the minimum absolute difference of the array [5,2,3,7,2] is |2 - 3| = 1. Note that it is not 0 because a[i] and a[j] must be different.

You are given an integer array nums and the array queries where queries[i] = [li, ri]. For each query i, compute the minimum absolute difference of the subarray nums[li...ri] containing the elements of nums between the 0-based indices li and ri (inclusive). Return an array ans where ans[i] is the answer to the ith query. A subarray is a contiguous sequence of elements in an array. The value of |x| is defined as:

x if x >= 0. -x if x < 0.

Example 1: Input: nums = [1,3,4,8], queries = [[0,1],[1,2],[2,3],[0,3]] Output: [2,1,4,1] Explanation: The queries are processed as follows: - queries[0] = [0,1]: The subarray is [1,3] and the minimum absolute difference is |1-3| = 2. - queries[1] = [1,2]: The subarray is [3,4] and the minimum absolute difference is |3-4| = 1. - queries[2] = [2,3]: The subarray is [4,8] and the minimum absolute difference is |4-8| = 4. - queries[3] = [0,3]: The subarray is [1,3,4,8] and the minimum absolute difference is |3-4| = 1.

Example 2: Input: nums = [4,5,2,2,7,10], queries = [[2,3],[0,2],[0,5],[3,5]] Output: [-1,1,1,3] Explanation: The queries are processed as follows: - queries[0] = [2,3]: The subarray is [2,2] and the minimum absolute difference is -1 because all the elements are the same. - queries[1] = [0,2]: The subarray is [4,5,2] and the minimum absolute difference is |4-5| = 1. - queries[2] = [0,5]: The subarray is [4,5,2,2,7,10] and the minimum absolute difference is |4-5| = 1. - queries[3] = [3,5]: The subarray is [2,7,10] and the minimum absolute difference is |7-10| = 3.

Constraints:

2 <= nums.length <= 105 1 <= nums[i] <= 100 1 <= queries.length <= 2 * 104 0 <= li < ri < nums.length

==
Problem Number: 1321 URL: https://leetcode.com/problems/remove-all-occurrences-of-a-substring Title: 1910. Remove All Occurrences of a Substring Problem Description: Given two strings s and part, perform the following operation on s until all occurrences of the substring part are removed:

Find the leftmost occurrence of the substring part and remove it from s.

Return s after removing all occurrences of part. A substring is a contiguous sequence of characters in a string. Example 1: Input: s = "daabcbaabcbc", part = "abc" Output: "dab" Explanation: The following operations are done: - s = "daabcbaabcbc", remove "abc" starting at index 2, so s = "dabaabcbc". - s = "dabaabcbc", remove "abc" starting at index 4, so s = "dababc". - s = "dababc", remove "abc" starting at index 3, so s = "dab". Now s has no occurrences of "abc".

Example 2: Input: s = "axxxxyyyyb", part = "xy" Output: "ab" Explanation: The following operations are done: - s = "axxxxyyyyb", remove "xy" starting at index 4 so s = "axxxyyyb". - s = "axxxyyyb", remove "xy" starting at index 3 so s = "axxyyb". - s = "axxyyb", remove "xy" starting at index 2 so s = "axyb". - s = "axyb", remove "xy" starting at index 1 so s = "ab". Now s has no occurrences of "xy".

Constraints:

1 <= s.length <= 1000 1 <= part.length <= 1000 s and part consists of lowercase English letters.

==
Problem Number: 1322 URL: https://leetcode.com/problems/maximum-alternating-subsequence-sum Title: 1911. Maximum Alternating Subsequence Sum Problem Description: The alternating sum of a 0-indexed array is defined as the sum of the elements at even indices minus the sum of the elements at odd indices.

For example, the alternating sum of [4,2,5,3] is (4 + 5) - (2 + 3) = 4.

Given an array nums, return the maximum alternating sum of any subsequence of nums (after reindexing the elements of the subsequence).

A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order. For example, [2,7,4] is a subsequence of [4,2,3,7,2,1,4] (the underlined elements), while [2,4,2] is not. Example 1: Input: nums = [4,2,5,3] Output: 7 Explanation: It is optimal to choose the subsequence [4,2,5] with alternating sum (4 + 5) - 2 = 7.

Example 2: Input: nums = [5,6,7,8] Output: 8 Explanation: It is optimal to choose the subsequence [8] with alternating sum 8.

Example 3: Input: nums = [6,2,1,2,4,5] Output: 10 Explanation: It is optimal to choose the subsequence [6,1,5] with alternating sum (6 + 5) - 1 = 10.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105

==
Problem Number: 1323 URL: https://leetcode.com/problems/cyclically-rotating-a-grid Title: 1914. Cyclically Rotating a Grid Problem Description: You are given an m x n integer matrix grid, where m and n are both even integers, and an integer k. The matrix is composed of several layers, which is shown in the below image, where each color is its own layer:

A cyclic rotation of the matrix is done by cyclically rotating each layer in the matrix. To cyclically rotate a layer once, each element in the layer will take the place of the adjacent element in the counter-clockwise direction. An example rotation is shown below:

Return the matrix after applying k cyclic rotations to it. Example 1:

Input: grid = [[40,10],[30,20]], k = 1 Output: [[10,20],[40,30]] Explanation: The figures above represent the grid at every state.

Example 2:

Input: grid = [[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]], k = 2 Output: [[3,4,8,12],[2,11,10,16],[1,7,6,15],[5,9,13,14]] Explanation: The figures above represent the grid at every state.

Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 50 Both m and n are even integers. 1 <= grid[i][j] <= 5000 1 <= k <= 109

==
Problem Number: 1324 URL: https://leetcode.com/problems/number-of-wonderful-substrings Title: 1915. Number of Wonderful Substrings Problem Description: A wonderful string is a string where at most one letter appears an odd number of times.

For example, ”ccjjc” and ”abab” are wonderful, but ”ab” is not.

Given a string word that consists of the first ten lowercase English letters (’a’ through ’j’), return the number of wonderful non-empty substrings in word. If the same substring appears multiple times in word, then count each occurrence separately. A substring is a contiguous sequence of characters in a string. Example 1: Input: word = ”aba” Output: 4 Explanation: The four wonderful substrings are underlined below: - ”aba” -> ”a” - ”aba” -> ”b” - ”aba” -> ”a” - ”aba” -> ”aba”

Example 2: Input: word = ”aabb” Output: 9 Explanation: The nine wonderful substrings are underlined below: - ”aabb” -> ”a” - ”aabb” -> ”aa” - ”aabb” -> ”aab” - ”aabb” -> ”aabb” - ”aabb” -> ”a” - ”aabb” -> ”abb” - ”aabb” -> ”b” - ”aabb” -> ”bb” - ”aabb” -> ”b”

Example 3: Input: word = ”he” Output: 2 Explanation: The two wonderful substrings are underlined below: - ”he” -> ”h” - ”he” -> ”e”

Constraints:

1 <= word.length <= 105 word consists of lowercase English letters from ’a’ to ’j’.

==

Problem Number: 1325 URL: https://leetcode.com/problems/eliminate-maximum-number-of-monsters Title: 1921. Eliminate Maximum Number of Monsters Problem Description: You are playing a video game where you are defending your city from a group of n monsters. You are given a 0-indexed integer array dist of size n, where dist[i] is the initial distance in kilometers of the ith monster from the city. The monsters walk toward the city at a constant speed. The speed of each monster is given to you in an integer array speed of size n, where speed[i] is the speed of the ith monster in kilometers per minute. You have a weapon that, once fully charged, can eliminate a single monster. However, the weapon takes one minute to charge.The weapon is fully charged at the very start. You lose when any monster reaches your city. If a monster reaches the city at the exact moment the weapon is fully charged, it counts as a loss, and the game ends before you can use your weapon. Return the maximum number of monsters that you can eliminate before you lose, or n if you can eliminate all the monsters before they reach the city. Example 1: Input: dist = [1,3,4], speed = [1,1,1] Output: 3 Explanation: In the beginning, the distances of the monsters are [1,3,4]. You eliminate the first monster. After a minute, the distances of the monsters are [X,2,3]. You eliminate the second monster. After a minute, the distances of the monsters are [X,X,2]. You eliminate the thrid monster. All 3 monsters can be eliminated. Example 2: Input: dist = [1,1,2,3], speed = [1,1,1,1] Output: 1 Explanation: In the beginning, the distances of the monsters are [1,1,2,3]. You eliminate the first monster. After a minute, the distances of the monsters are [X,0,1,2], so you lose. You can only eliminate 1 monster.

Example 3: Input: dist = [3,2,4], speed = [5,3,2] Output: 1 Explanation: In the beginning, the distances of the monsters are [3,2,4]. You eliminate the first monster. After a minute, the distances of the monsters are [X,0,2], so you lose. You can only eliminate 1 monster.

Constraints:

n == dist.length == speed.length 1 <= n <= 105 1 <= dist[i], speed[i] <= 105

==
Problem Number: 1326 URL: https://leetcode.com/problems/count-good-numbers Title: 1922. Count Good Numbers Problem Description: A digit string is good if the digits (0-indexed) at even indices are even and the digits at odd indices are prime (2, 3, 5, or 7).

For example, "2582" is good because the digits (2 and 8) at even positions are even and the digits (5 and 2) at odd positions are prime. However, "3245" is not good because 3 is at an even index but is not even.

Given an integer n, return the total number of good digit strings of length n. Since the answer may be large, return it modulo 109 + 7. A digit string is a string consisting of digits 0 through 9 that may contain leading zeros. Example 1: Input: n = 1 Output: 5 Explanation: The good numbers of length 1 are "0", "2", "4", "6", "8".

Example 2: Input: n = 4 Output: 400

Example 3: Input: n = 50 Output: 564908303

Constraints:

1 <= n <= 1015

==
Problem Number: 1327 URL: https://leetcode.com/problems/nearest-exit-from-entrance-in-maze Title: 1926. Nearest Exit from Entrance in Maze Problem Description: You are given an m x n matrix maze (0-indexed) with empty cells (represented as '.') and walls (represented as '+'). You are also given the entrance of the maze, where entrance = [entrancerow, entrancecol] denotes the row and column of the cell you are initially standing at. In one step, you can move one cell up, down, left, or right. You cannot step into a cell with a wall, and you cannot step outside the maze. Your goal is to find the nearest exit from the entrance. An exit is defined as an empty cell that is at the border of the maze. The entrance does not count as an exit. Return the number of steps in the shortest path from the entrance to the nearest exit, or -1 if no such path exists. Example 1:

Input: maze = [["+","+",".","+"],[".",".",".","+"],["+","+","+","."]], entrance = [1,2] Output: 1 Explanation: There are 3 exits in this maze at [1,0], [0,2], and [2,3]. Initially, you are at the entrance cell [1,2]. - You can reach [1,0] by moving

2 steps left. - You can reach [0,2] by moving 1 step up. It is impossible to reach [2,3] from the entrance. Thus, the nearest exit is [0,2], which is 1 step away.

Example 2:

Input: maze = [["+","+","+"],[".",".","."],["+","+","+"]], entrance = [1,0] Output: 2 Explanation: There is 1 exit in this maze at [1,2]. [1,0] does not count as an exit since it is the entrance cell. Initially, you are at the entrance cell [1,0]. - You can reach [1,2] by moving 2 steps right. Thus, the nearest exit is [1,2], which is 2 steps away.

Example 3:

Input: maze = [[".","+"]], entrance = [0,0] Output: -1 Explanation: There are no exits in this maze.

Constraints:

maze.length == m maze[i].length == n 1 <= m, n <= 100 maze[i][j] is either '.' or '+'. entrance.length == 2 0 <= entrancerow < m 0 <= entrancecol < n entrance will always be an empty cell.

==

Problem Number: 1328 URL: https://leetcode.com/problems/sum-game Title: 1927. Sum Game Problem Description: Alice and Bob take turns playing a game, with Alice starting first. You are given a string num of even length consisting of digits and '?' characters. On each turn, a player will do the following if there is still at least one '?' in num:

Choose an index i where num[i] == '?'. Replace num[i] with any digit between '0' and '9'.

The game ends when there are no more '?' characters in num. For Bob to win, the sum of the digits in the first half of num must be equal to the sum of the digits in the second half. For Alice to win, the sums must not be equal.

For example, if the game ended with num = "243801", then Bob wins because 2+4+3 = 8+0+1. If the game ended with num = "243803", then Alice wins because 2+4+3 != 8+0+3.

Assuming Alice and Bob play optimally, return true if Alice will win and false if Bob will win. Example 1: Input: num = "5023" Output: false Explanation: There are no moves to be made. The sum of the first half is equal to the sum of the second half: 5 + 0 = 2 + 3.

Example 2: Input: num = "25??" Output: true Explanation: Alice can replace one of the '?'s with '9' and it will be impossible for Bob to make the sums equal.

Example 3: Input: num = "?3295???" Output: false Explanation: It can be proven that Bob will always win. One possible outcome is: - Alice replaces the first '?' with '9'. num = "93295???". - Bob replaces one of the '?' in the right half with '9'. num = "932959??". - Alice replaces one of the '?' in the right half with '2'. num = "9329592?". - Bob replaces the last '?' in the right half with '7'. num = "93295927". Bob wins because 9 + 3 + 2 + 9 = 5 + 9 + 2 + 7.

Constraints:

2 <= num.length <= 105 num.length is even. num consists of only digits and '?'.

==

Problem Number: 1329 URL: https://leetcode.com/problems/unique-length-3-palindromic-subsequences Title: 1930. Unique Length-3 Palindromic Subsequences Problem Description: Given a string s, return the number of unique palindromes of length three that are a subsequence of s. Note that even if there are multiple ways to obtain the same subsequence, it is still only counted once. A palindrome is a string that reads the same forwards and backwards. A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, "ace" is a subsequence of "abcde".

Example 1: Input: s = "aabca" Output: 3 Explanation: The 3 palindromic subsequences of length 3 are: - "aba" (subsequence of "aabca") - "aaa" (subsequence of "aabca") - "aca" (subsequence of "aabca")

Example 2: Input: s = "adc" Output: 0 Explanation: There are no palindromic subsequences of length 3 in "adc".

Example 3: Input: s = "bbcbaba" Output: 4 Explanation: The 4 palindromic subsequences of length 3 are: - "bbb" (subsequence of "bbcbaba") - "bcb" (subsequence of "bbcbaba") - "bab" (subsequence of "bbcbaba") - "aba" (subsequence of "bbcbaba")

Constraints:

3 <= s.length <= 105 s consists of only lowercase English letters.

==

Problem Number: 1330 URL: https://leetcode.com/problems/add-minimum-number-of-rungs Title: 1936. Add Minimum Number of Rungs Problem Description: You are given a strictly increasing integer array rungs that represents the height of rungs on a ladder. You are currently on the floor at height 0, and you want to reach the last rung. You are also given an integer dist. You can only climb to the next highest rung if the distance between where you are currently at (the floor or on a rung) and the next rung is at most dist. You are able to insert rungs at any positive integer height if a rung is not already there. Return the minimum number of rungs that must be added to the ladder in order for you to climb to the last rung. Example 1: Input: rungs = [1,3,5,10], dist = 2 Output: 2 Explanation: You currently cannot reach the last rung. Add rungs at heights 7 and 8 to climb this ladder. The ladder will now have rungs at [1,3,5,7,8,10].

Example 2: Input: rungs = [3,6,8,10], dist = 3 Output: 0 Explanation: This ladder can be climbed without adding additional rungs.

Example 3: Input: rungs = [3,4,6,7], dist = 2 Output: 1 Explanation: You currently cannot reach the first rung from the ground. Add a rung at height 1 to climb this ladder. The ladder will now have rungs at [1,3,4,6,7].

Constraints:

1 <= rungs.length <= 105 1 <= rungs[i] <= 109 1 <= dist <= 109 rungs is strictly increasing.

==
Problem Number: 1331 URL: https://leetcode.com/problems/maximum-number-of-points-with-cost Title: 1937. Maximum Number of Points with Cost Problem Description: You are given an m x n integer matrix points (0-indexed). Starting with 0 points, you want to maximize the number of points you can get from the matrix. To gain points, you must pick one cell in each row. Picking the cell at coordinates (r, c) will add points[r][c] to your score. However, you will lose points if you pick a cell too far from the cell that you picked in the previous row. For every two adjacent rows r and r + 1 (where 0 <= r < m - 1), picking cells at coordinates (r, c1) and (r + 1, c2) will subtract abs(c1 - c2) from your score. Return the maximum number of points you can achieve. abs(x) is defined as:

x for x >= 0. -x for x < 0.

Example 1:

Input: points = [[1,2,3],[1,5,1],[3,1,1]] Output: 9 Explanation: The blue cells denote the optimal cells to pick, which have coordinates (0, 2), (1, 1), and (2, 0). You add 3 + 5 + 3 = 11 to your score. However, you must subtract abs(2 - 1) + abs(1 - 0) = 2 from your score. Your final score is 11 - 2 = 9.

Example 2:

Input: points = [[1,5],[2,3],[4,2]] Output: 11 Explanation: The blue cells denote the optimal cells to pick, which have coordinates (0, 1), (1, 1), and (2, 0). You add 5 + 3 + 4 = 12 to your score. However, you must subtract abs(1 - 1) + abs(1 - 0) = 1 from your score. Your final score is 12 - 1 = 11.

Constraints:

m == points.length n == points[r].length 1 <= m, n <= 105 1 <= m * n <= 105 0 <= points[r][c] <= 105

==
Problem Number: 1332 URL: https://leetcode.com/problems/the-number-of-the-smallest-unoccupied-chair Title: 1942. The Number of the Smallest Unoccupied Chair Problem Description: There is a party where n friends numbered from 0 to n - 1 are attending. There is an infinite number of chairs

in this party that are numbered from 0 to infinity. When a friend arrives at the party, they sit on the unoccupied chair with the smallest number.

For example, if chairs 0, 1, and 5 are occupied when a friend comes, they will sit on chair number 2.

When a friend leaves the party, their chair becomes unoccupied at the moment they leave. If another friend arrives at that same moment, they can sit in that chair. You are given a 0-indexed 2D integer array times where times[i] = [arrivali, leavingi], indicating the arrival and leaving times of the ith friend respectively, and an integer targetFriend. All arrival times are distinct. Return the chair number that the friend numbered targetFriend will sit on. Example 1: Input: times = [[1,4],[2,3],[4,6]], targetFriend = 1 Output: 1 Explanation: - Friend 0 arrives at time 1 and sits on chair 0. - Friend 1 arrives at time 2 and sits on chair 1. - Friend 1 leaves at time 3 and chair 1 becomes empty. - Friend 0 leaves at time 4 and chair 0 becomes empty. - Friend 2 arrives at time 4 and sits on chair 0. Since friend 1 sat on chair 1, we return 1.

Example 2: Input: times = [[3,10],[1,5],[2,6]], targetFriend = 0 Output: 2 Explanation: - Friend 1 arrives at time 1 and sits on chair 0. - Friend 2 arrives at time 2 and sits on chair 1. - Friend 0 arrives at time 3 and sits on chair 2. - Friend 1 leaves at time 5 and chair 0 becomes empty. - Friend 2 leaves at time 6 and chair 1 becomes empty. - Friend 0 leaves at time 10 and chair 2 becomes empty. Since friend 0 sat on chair 2, we return 2.

Constraints:

n == times.length 2 <= n <= 104 times[i].length == 2 1 <= arrivali < leavingi <= 105 0 <= targetFriend <= n - 1 Each arrivali time is distinct.

==

Problem Number: 1333 URL: https://leetcode.com/problems/describe-the-painting Title: 1943. Describe the Painting Problem Description: There is a long and thin painting that can be represented by a number line. The painting was painted with multiple overlapping segments where each segment was painted with a unique color. You are given a 2D integer array segments, where segments[i] = [starti, endi, colori] represents the half-closed segment [starti, endi) with colori as the color. The colors in the overlapping segments of the painting were mixed when it was painted. When two or more colors mix, they form a new color that can be represented as a set of mixed colors.

For example, if colors 2, 4, and 6 are mixed, then the resulting mixed color is {2,4,6}.

For the sake of simplicity, you should only output the sum of the elements in the set rather than the full set. You want to describe the painting with the minimum number of non-overlapping half-closed segments of these mixed colors. These segments can be represented by the 2D array painting where painting[j] = [leftj, rightj, mixj] describes a half-closed segment [leftj, rightj) with the mixed color sum of mixj.

For example, the painting created with segments = [[1,4,5],[1,7,7]] can be described by painting = [[1,4,12],[4,7,7]] because:

[1,4) is colored {5,7} (with a sum of 12) from both the first and second segments. [4,7) is colored {7} from only the second segment.

Return the 2D array painting describing the finished painting (excluding any parts that are not painted). You may return the segments in any order. A half-closed segment [a, b) is the section of the number line between points a and b including point a and not including point b.   Example 1:

Input: segments = [[1,4,5],[4,7,7],[1,7,9]] Output: [[1,4,14],[4,7,16]] Explanation: The painting can be described as follows: - [1,4) is colored {5,9} (with a sum of 14) from the first and third segments. - [4,7) is colored {7,9} (with a sum of 16) from the second and third segments.

Example 2:

Input: segments = [[1,7,9],[6,8,15],[8,10,7]] Output: [[1,6,9],[6,7,24],[7,8,15],[8,10,7]] Explanation: The painting can be described as follows: - [1,6) is colored 9 from the first segment. - [6,7) is colored {9,15} (with a sum of 24) from the first and second segments. - [7,8) is colored 15 from the second segment. - [8,10) is colored 7 from the third segment.

Example 3:

Input: segments = [[1,4,5],[1,4,7],[4,7,1],[4,7,11]] Output: [[1,4,12],[4,7,12]] Explanation: The painting can be described as follows: - [1,4) is colored {5,7} (with a sum of 12) from the first and second segments. - [4,7) is colored {1,11} (with a sum of 12) from the third and fourth segments. Note that returning a single segment [1,7) is incorrect because the mixed color sets are different.

Constraints:

1 <= segments.length <= 2 * 104 segments[i].length == 3 1 <= starti < endi <= 105 1 <= colori <= 109 Each colori is distinct.

==
Problem Number: 1334 URL: https://leetcode.com/problems/largest-number-after-mutating-substring Title: 1946. Largest Number After Mutating Substring Problem Description: You are given a string num, which represents a large integer. You are also given a 0-indexed integer array change of length 10 that maps each digit 0-9 to another digit. More formally, digit d maps to digit change[d]. You may choose to mutate a single substring of num. To mutate a substring, replace each digit num[i] with the digit it maps to in change (i.e. replace num[i] with change[num[i]]). Return a string representing the largest possible integer after mutating (or choosing not to) a single substring of num. A substring is a contiguous sequence of characters within the string.   Example 1: Input: num = ”132”, change = [9,8,5,0,3,6,4,2,6,8] Output: ”832” Explanation: Replace the substring ”1”: - 1 maps to change[1] = 8. Thus, ”132” becomes ”832”. ”832” is the largest number that can be created, so return it.

Example 2: Input: num = "021", change = [9,4,3,5,7,2,1,9,0,6] Output: "934" Explanation: Replace the substring "021": - 0 maps to change[0] = 9. - 2 maps to change[2] = 3. - 1 maps to change[1] = 4. Thus, "021" becomes "934". "934" is the largest number that can be created, so return it.

Example 3: Input: num = "5", change = [1,4,7,5,3,2,5,6,9,4] Output: "5" Explanation: "5" is already the largest number that can be created, so return it.

Constraints:

1 <= num.length <= 105 num consists of only digits 0-9. change.length == 10 0 <= change[d] <= 9

==
Problem Number: 1335 URL: https://leetcode.com/problems/maximum-compatibility-score-sum Title: 1947. Maximum Compatibility Score Sum Problem Description: There is a survey that consists of n questions where each question's answer is either 0 (no) or 1 (yes). The survey was given to m students numbered from 0 to m - 1 and m mentors numbered from 0 to m - 1. The answers of the students are represented by a 2D integer array students where students[i] is an integer array that contains the answers of the ith student (0-indexed). The answers of the mentors are represented by a 2D integer array mentors where mentors[j] is an integer array that contains the answers of the jth mentor (0-indexed). Each student will be assigned to one mentor, and each mentor will have one student assigned to them. The compatibility score of a student-mentor pair is the number of answers that are the same for both the student and the mentor.

For example, if the student's answers were [1, 0, 1] and the mentor's answers were [0, 0, 1], then their compatibility score is 2 because only the second and the third answers are the same.

You are tasked with finding the optimal student-mentor pairings to maximize the sum of the compatibility scores. Given students and mentors, return the maximum compatibility score sum that can be achieved. Example 1: Input: students = [[1,1,0],[1,0,1],[0,0,1]], mentors = [[1,0,0],[0,0,1],[1,1,0]] Output: 8 Explanation: We assign students to mentors in the following way: - student 0 to mentor 2 with a compatibility score of 3. - student 1 to mentor 0 with a compatibility score of 2. - student 2 to mentor 1 with a compatibility score of 3. The compatibility score sum is 3 + 2 + 3 = 8.

Example 2: Input: students = [[0,0],[0,0],[0,0]], mentors = [[1,1],[1,1],[1,1]] Output: 0 Explanation: The compatibility score of any student-mentor pair is 0.

Constraints:

m == students.length == mentors.length n == students[i].length == mentors[j].length 1 <= m, n <= 8 students[i][k] is either 0 or 1. mentors[j][k] is either 0 or 1.

==

Problem Number: 1336 URL: https://leetcode.com/problems/maximum-number-of-weeks-for-which-you-can-work Title: 1953. Maximum Number of Weeks for Which You Can Work Problem Description: There are n projects numbered from 0 to n - 1. You are given an integer array milestones where each milestones[i] denotes the number of milestones the ith project has. You can work on the projects following these two rules:

Every week, you will finish exactly one milestone of one project. You must work every week. You cannot work on two milestones from the same project for two consecutive weeks.

Once all the milestones of all the projects are finished, or if the only milestones that you can work on will cause you to violate the above rules, you will stop working. Note that you may not be able to finish every project's milestones due to these constraints. Return the maximum number of weeks you would be able to work on the projects without violating the rules mentioned above. Example 1: Input: milestones = [1,2,3] Output: 6 Explanation: One possible scenario is: - During the 1st week, you will work on a milestone of project 0. - During the 2nd week, you will work on a milestone of project 2. - During the 3rd week, you will work on a milestone of project 1. - During the 4th week, you will work on a milestone of project 2. - During the 5th week, you will work on a milestone of project 1. - During the 6th week, you will work on a milestone of project 2. The total number of weeks is 6.

Example 2: Input: milestones = [5,2,1] Output: 7 Explanation: One possible scenario is: - During the 1st week, you will work on a milestone of project 0. - During the 2nd week, you will work on a milestone of project 1. - During the 3rd week, you will work on a milestone of project 0. - During the 4th week, you will work on a milestone of project 1. - During the 5th week, you will work on a milestone of project 0. - During the 6th week, you will work on a milestone of project 2. - During the 7th week, you will work on a milestone of project 0. The total number of weeks is 7. Note that you cannot work on the last milestone of project 0 on 8th week because it would violate the rules. Thus, one milestone in project 0 will remain unfinished.

Constraints:

n == milestones.length 1 <= n <= 105 1 <= milestones[i] <= 109

==

Problem Number: 1337 URL: https://leetcode.com/problems/minimum-garden-perimeter-to-collect-enough-apples Title: 1954. Minimum Garden Perimeter to Collect Enough Apples Problem Description: In a garden represented as an infinite 2D grid, there is an apple tree planted at every integer coordinate. The apple tree planted at an integer coordinate (i, j) has |i| + |j| apples growing on it. You will buy an axis-aligned square plot of land that is centered at (0, 0). Given an integer neededApples, return the minimum perimeter of a plot such that at least neededApples apples are inside or on the perimeter of that plot. The value of |x| is defined as:

x if x >= 0 -x if x < 0

Example 1:

Input: neededApples = 1 Output: 8 Explanation: A square plot of side length 1 does not contain any apples. However, a square plot of side length 2 has 12 apples inside (as depicted in the image above). The perimeter is 2 * 4 = 8.

Example 2: Input: neededApples = 13 Output: 16

Example 3: Input: neededApples = 1000000000 Output: 5040

Constraints:

1 <= neededApples <= 1015

==
Problem Number: 1338 URL: https://leetcode.com/problems/check-if-move-is-legal Title: 1958. Check if Move is Legal Problem Description: You are given a 0-indexed 8 x 8 grid board, where board[r][c] represents the cell (r, c) on a game board. On the board, free cells are represented by '.', white cells are represented by 'W', and black cells are represented by 'B'. Each move in this game consists of choosing a free cell and changing it to the color you are playing as (either white or black). However, a move is only legal if, after changing it, the cell becomes the endpoint of a good line (horizontal, vertical, or diagonal). A good line is a line of three or more cells (including the endpoints) where the endpoints of the line are one color, and the remaining cells in the middle are the opposite color (no cells in the line are free). You can find examples for good lines in the figure below:

Given two integers rMove and cMove and a character color representing the color you are playing as (white or black), return true if changing cell (rMove, cMove) to color color is a legal move, or false if it is not legal. Example 1:

Input: board = [[".",".",".","B",".",".",".","."],[".",".",".","W",".",".",".","."],[".",".",".","W",".",".",".","."],[".",".",".","W","."
rMove = 4, cMove = 3, color = "B" Output: true Explanation: '.', 'W', and 'B' are represented by the colors blue, white, and black respectively, and cell (rMove, cMove) is marked with an 'X'. The two good lines with the chosen cell as an endpoint are annotated above with the red rectangles.

Example 2:

Input: board = [[".",".",".",".",".",".",".","."],[".","B",".",".","W",".",".","."],[".",".","W",".",".",".",".","."],[".",".",".","W","B
rMove = 4, cMove = 4, color = "W" Output: false Explanation: While there are good lines with the chosen cell as a middle cell, there are no good lines with the chosen cell as an endpoint.

Constraints:

board.length == board[r].length == 8 0 <= rMove, cMove < 8 board[rMove][cMove] == '.' color is either 'B' or 'W'.

==

Problem Number: 1339 URL: https://leetcode.com/problems/minimum-total-space-wasted-with-k-resizing-operations Title: 1959. Minimum Total Space Wasted With K Resizing Operations Problem Description: You are currently designing a dynamic array. You are given a 0-indexed integer array nums, where nums[i] is the number of elements that will be in the array at time i. In addition, you are given an integer k, the maximum number of times you can resize the array (to any size). The size of the array at time t, sizet, must be at least nums[t] because there needs to be enough space in the array to hold all the elements. The space wasted at time t is defined as sizet - nums[t], and the total space wasted is the sum of the space wasted across every time t where 0 <= t < nums.length. Return the minimum total space wasted if you can resize the array at most k times. Note: The array can have any size at the start and does not count towards the number of resizing operations. Example 1: Input: nums = [10,20], k = 0 Output: 10 Explanation: size = [20,20]. We can set the initial size to be 20. The total wasted space is (20 - 10) + (20 - 20) = 10.

Example 2: Input: nums = [10,20,30], k = 1 Output: 10 Explanation: size = [20,20,30]. We can set the initial size to be 20 and resize to 30 at time 2. The total wasted space is (20 - 10) + (20 - 20) + (30 - 30) = 10.

Example 3: Input: nums = [10,20,15,30,20], k = 2 Output: 15 Explanation: size = [10,20,20,30,30]. We can set the initial size to 10, resize to 20 at time 1, and resize to 30 at time 3. The total wasted space is (10 - 10) + (20 - 20) + (20 - 15) + (30 - 30) + (30 - 20) = 15.

Constraints:

1 <= nums.length <= 200 1 <= nums[i] <= 106 0 <= k <= nums.length - 1

==

Problem Number: 1340 URL: https://leetcode.com/problems/remove-stones-to-minimize-the-total Title: 1962. Remove Stones to Minimize the Total Problem Description: You are given a 0-indexed integer array piles, where piles[i] represents the number of stones in the ith pile, and an integer k. You should apply the following operation exactly k times:

Choose any piles[i] and remove floor(piles[i] / 2) stones from it.

Notice that you can apply the operation on the same pile more than once. Return the minimum possible total number of stones remaining after applying the k operations. floor(x) is the greatest integer that is smaller than or equal to x (i.e., rounds x down). Example 1: Input: piles = [5,4,9], k = 2 Output: 12 Explanation: Steps of a possible scenario are: - Apply the operation on pile 2. The resulting piles are [5,4,5]. - Apply the operation on pile 0. The resulting piles are [3,4,5]. The total number of stones in [3,4,5] is 12.

Example 2: Input: piles = [4,3,6,7], k = 3 Output: 12 Explanation: Steps of a possible scenario are: - Apply the operation on pile 2. The resulting piles are [4,3,3,7]. - Apply the operation on pile 3. The resulting piles are [4,3,3,4]. - Apply the operation on pile 0. The resulting piles are [2,3,3,4]. The total number of stones in [2,3,3,4] is 12.

Constraints:

1 <= piles.length <= 105 1 <= piles[i] <= 104 1 <= k <= 105

==

Problem Number: 1341 URL: https://leetcode.com/problems/minimum-number-of-swaps-to-make-the-string-balanced Title: 1963. Minimum Number of Swaps to Make the String Balanced Problem Description: You are given a 0-indexed string s of even length n. The string consists of exactly n / 2 opening brackets '[' and n / 2 closing brackets ']'. A string is called balanced if and only if:

It is the empty string, or It can be written as AB, where both A and B are balanced strings, or It can be written as [C], where C is a balanced string.

You may swap the brackets at any two indices any number of times. Return the minimum number of swaps to make s balanced. Example 1: Input: s = "][][" Output: 1 Explanation: You can make the string balanced by swapping index 0 with index 3. The resulting string is "[[]]".

Example 2: Input: s = "]]][[[" Output: 2 Explanation: You can do the following to make the string balanced: - Swap index 0 with index 4. s = "[]][][". - Swap index 1 with index 5. s = "[[][]]". The resulting string is "[[][]]".

Example 3: Input: s = "[]" Output: 0 Explanation: The string is already balanced.

Constraints:

n == s.length 2 <= n <= 106 n is even. s[i] is either '[' or ']'. The number of opening brackets '[' equals n / 2, and the number of closing brackets ']' equals n / 2.

==

Problem Number: 1342 URL: https://leetcode.com/problems/array-with-elements-not-equal-to-average-of-neighbors Title: 1968. Array With Elements Not Equal to Average of Neighbors Problem Description: You are given a 0-indexed array nums of distinct integers. You want to rearrange the elements in the array such that every element in the rearranged array is not equal to the average of its neighbors. More formally, the rearranged array should have the property such that for every i in the range 1 <= i < nums.length - 1, (nums[i-1] + nums[i+1]) / 2 is not equal to nums[i]. Return any rearrangement of nums that meets the requirements. Example 1: Input: nums = [1,2,3,4,5] Output: [1,2,4,5,3] Explanation: When i=1, nums[i] = 2, and the average of its neighbors is (1+4) / 2 = 2.5. When i=2, nums[i] = 4, and the average of its neighbors is (2+5) / 2 = 3.5. When i=3, nums[i] = 5, and the average of its neighbors is (4+3) / 2 = 3.5.

Example 2: Input: nums = [6,2,0,9,7] Output: [9,7,6,2,0] Explanation: When i=1, nums[i] = 7, and the average of its neighbors is (9+6) / 2 = 7.5. When i=2, nums[i] = 6, and the average of its neighbors is (7+2) / 2 = 4.5. When i=3, nums[i] = 2, and the average of its neighbors is (6+0) / 2 = 3.

Constraints:

3 <= nums.length <= 105 0 <= nums[i] <= 105

==
Problem Number: 1343 URL: https://leetcode.com/problems/minimum-non-zero-product-of-the-array-elements Title: 1969. Minimum Non-Zero Product of the Array Elements Problem Description: You are given a positive integer p. Consider an array nums (1-indexed) that consists of the integers in the inclusive range [1, 2p - 1] in their binary representations. You are allowed to do the following operation any number of times:

Choose two elements x and y from nums. Choose a bit in x and swap it with its corresponding bit in y. Corresponding bit refers to the bit that is in the same position in the other integer.

For example, if x = 1101 and y = 0011, after swapping the 2nd bit from the right, we have x = 1111 and y = 0001. Find the minimum non-zero product of nums after performing the above operation any number of times. Return this product modulo 109 + 7. Note: The answer should be the minimum product before the modulo operation is done. Example 1: Input: p = 1 Output: 1 Explanation: nums = [1]. There is only one element, so the product equals that element.

Example 2: Input: p = 2 Output: 6 Explanation: nums = [01, 10, 11]. Any swap would either make the product 0 or stay the same. Thus, the array product of 1 * 2 * 3 = 6 is already minimized.

Example 3: Input: p = 3 Output: 1512 Explanation: nums = [001, 010, 011, 100, 101, 110, 111] - In the first operation we can swap the leftmost bit of the second and fifth elements. - The resulting array is [001, 110, 011, 100, 001, 110, 111]. - In the second operation we can swap the middle bit of the third and fourth elements. - The resulting array is [001, 110, 001, 110, 001, 110, 111]. The array product is 1 * 6 * 1 * 6 * 1 * 6 * 7 = 1512, which is the minimum possible product.

Constraints:

1 <= p <= 60

==
Problem Number: 1344 URL: https://leetcode.com/problems/maximum-matrix-sum Title: 1975. Maximum Matrix Sum Problem Description: You are

given an n x n integer matrix. You can do the following operation any number of times:

Choose any two adjacent elements of matrix and multiply each of them by -1.

Two elements are considered adjacent if and only if they share a border. Your goal is to maximize the summation of the matrix's elements. Return the maximum sum of the matrix's elements using the operation mentioned above. Example 1:

Input: matrix = [[1,-1],[-1,1]] Output: 4 Explanation: We can follow the following steps to reach sum equals 4: - Multiply the 2 elements in the first row by -1. - Multiply the 2 elements in the first column by -1.

Example 2:

Input: matrix = [[1,2,3],[-1,-2,-3],[1,2,3]] Output: 16 Explanation: We can follow the following step to reach sum equals 16: - Multiply the 2 last elements in the second row by -1.

Constraints:

n == matrix.length == matrix[i].length 2 <= n <= 250 -105 <= matrix[i][j] <= 105

==

Problem Number: 1345 URL: https://leetcode.com/problems/number-of-ways-to-arrive-at-destination Title: 1976. Number of Ways to Arrive at Destination Problem Description: You are in a city that consists of n intersections numbered from 0 to n - 1 with bi-directional roads between some intersections. The inputs are generated such that you can reach any intersection from any other intersection and that there is at most one road between any two intersections. You are given an integer n and a 2D integer array roads where roads[i] = [ui, vi, timei] means that there is a road between intersections ui and vi that takes timei minutes to travel. You want to know in how many ways you can travel from intersection 0 to intersection n - 1 in the shortest amount of time. Return the number of ways you can arrive at your destination in the shortest amount of time. Since the answer may be large, return it modulo 109 + 7. Example 1:

Input: n = 7, roads = [[0,6,7],[0,1,2],[1,2,3],[1,3,3],[6,3,3],[3,5,1],[6,5,1],[2,5,1],[0,4,5],[4,6,2]] Output: 4 Explanation: The shortest amount of time it takes to go from intersection 0 to intersection 6 is 7 minutes. The four ways to get there in 7 minutes are: - 0 6 - 0 4 6 - 0 1 2 5 6 - 0 1 3 5 6

Example 2: Input: n = 2, roads = [[1,0,10]] Output: 1 Explanation: There is only one way to go from intersection 0 to intersection 1, and it takes 10 minutes.

Constraints:

1 <= n <= 200 n - 1 <= roads.length <= n * (n - 1) / 2 roads[i].length == 3 0 <= ui, vi <= n - 1 1 <= timei <= 109 ui != vi There is at most one road

connecting any two intersections. You can reach any intersection from any other intersection.

==
Problem Number: 1346 URL: https://leetcode.com/problems/find-unique-binary-string Title: 1980. Find Unique Binary String Problem Description: Given an array of strings nums containing n unique binary strings each of length n, return a binary string of length n that does not appear in nums. If there are multiple answers, you may return any of them. Example 1: Input: nums = ["01","10"] Output: "11" Explanation: "11" does not appear in nums. "00" would also be correct.

Example 2: Input: nums = ["00","01"] Output: "11" Explanation: "11" does not appear in nums. "10" would also be correct.

Example 3: Input: nums = ["111","011","001"] Output: "101" Explanation: "101" does not appear in nums. "000", "010", "100", and "110" would also be correct.

Constraints:

n == nums.length 1 <= n <= 16 nums[i].length == n nums[i] is either '0' or '1'. All the strings of nums are unique.

==
Problem Number: 1347 URL: https://leetcode.com/problems/minimize-the-difference-between-target-and-chosen-elements Title: 1981. Minimize the Difference Between Target and Chosen Elements Problem Description: You are given an m x n integer matrix mat and an integer target. Choose one integer from each row in the matrix such that the absolute difference between target and the sum of the chosen elements is minimized. Return the minimum absolute difference. The absolute difference between two numbers a and b is the absolute value of a - b. Example 1:

Input: mat = [[1,2,3],[4,5,6],[7,8,9]], target = 13 Output: 0 Explanation: One possible choice is to: - Choose 1 from the first row. - Choose 5 from the second row. - Choose 7 from the third row. The sum of the chosen elements is 13, which equals the target, so the absolute difference is 0.

Example 2:

Input: mat = [[1],[2],[3]], target = 100 Output: 94 Explanation: The best possible choice is to: - Choose 1 from the first row. - Choose 2 from the second row. - Choose 3 from the third row. The sum of the chosen elements is 6, and the absolute difference is 94.

Example 3:

Input: mat = [[1,2,9,8,7]], target = 6 Output: 1 Explanation: The best choice is to choose 7 from the first row. The absolute difference is 1.

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 70 1 <= mat[i][j] <= 70 1 <= target <= 800

===

Problem Number: 1348 URL: https://leetcode.com/problems/find-the-kth-largest-integer-in-the-array Title: 1985. Find the Kth Largest Integer in the Array Problem Description: You are given an array of strings nums and an integer k. Each string in nums represents an integer without leading zeros. Return the string that represents the kth largest integer in nums. Note: Duplicate numbers should be counted distinctly. For example, if nums is ["1","2","2"], "2" is the first largest integer, "2" is the second-largest integer, and "1" is the third-largest integer. Example 1: Input: nums = ["3","6","7","10"], k = 4 Output: "3" Explanation: The numbers in nums sorted in non-decreasing order are ["3","6","7","10"]. The 4th largest integer in nums is "3".

Example 2: Input: nums = ["2","21","12","1"], k = 3 Output: "2" Explanation: The numbers in nums sorted in non-decreasing order are ["1","2","12","21"]. The 3rd largest integer in nums is "2".

Example 3: Input: nums = ["0","0"], k = 2 Output: "0" Explanation: The numbers in nums sorted in non-decreasing order are ["0","0"]. The 2nd largest integer in nums is "0".

Constraints:

1 <= k <= nums.length <= 104 1 <= nums[i].length <= 100 nums[i] consists of only digits. nums[i] will not have any leading zeros.

===

Problem Number: 1349 URL: https://leetcode.com/problems/minimum-number-of-work-sessions-to-finish-the-tasks Title: 1986. Minimum Number of Work Sessions to Finish the Tasks Problem Description: There are n tasks assigned to you. The task times are represented as an integer array tasks of length n, where the ith task takes tasks[i] hours to finish. A work session is when you work for at most sessionTime consecutive hours and then take a break. You should finish the given tasks in a way that satisfies the following conditions:

If you start a task in a work session, you must complete it in the same work session. You can start a new task immediately after finishing the previous one. You may complete the tasks in any order.

Given tasks and sessionTime, return the minimum number of work sessions needed to finish all the tasks following the conditions above. The tests are generated such that sessionTime is greater than or equal to the maximum element in tasks[i]. Example 1: Input: tasks = [1,2,3], sessionTime = 3 Output: 2 Explanation: You can finish the tasks in two work sessions. - First work session: finish the first and the second tasks in 1 + 2 = 3 hours. - Second work session: finish the third task in 3 hours.

Example 2: Input: tasks = [3,1,3,1,1], sessionTime = 8 Output: 2 Explanation: You can finish the tasks in two work sessions. - First work session: finish all the tasks except the last one in 3 + 1 + 3 + 1 = 8 hours. - Second work session: finish the last task in 1 hour.

Example 3: Input: tasks = [1,2,3,4,5], sessionTime = 15 Output: 1 Explanation: You can finish all the tasks in one work session.

Constraints:

n == tasks.length 1 <= n <= 14 1 <= tasks[i] <= 10 max(tasks[i]) <= sessionTime <= 15

==
Problem Number: 1350 URL: https://leetcode.com/problems/find-all-groups-of-farmland Title: 1992. Find All Groups of Farmland Problem Description: You are given a 0-indexed m x n binary matrix land where a 0 represents a hectare of forested land and a 1 represents a hectare of farmland. To keep the land organized, there are designated rectangular areas of hectares that consist entirely of farmland. These rectangular areas are called groups. No two groups are adjacent, meaning farmland in one group is not four-directionally adjacent to another farmland in a different group. land can be represented by a coordinate system where the top left corner of land is (0, 0) and the bottom right corner of land is (m-1, n-1). Find the coordinates of the top left and bottom right corner of each group of farmland. A group of farmland with a top left corner at (r1, c1) and a bottom right corner at (r2, c2) is represented by the 4-length array [r1, c1, r2, c2]. Return a 2D array containing the 4-length arrays described above for each group of farmland in land. If there are no groups of farmland, return an empty array. You may return the answer in any order. Example 1:

Input: land = [[1,0,0],[0,1,1],[0,1,1]] Output: [[0,0,0,0],[1,1,2,2]] Explanation: The first group has a top left corner at land[0][0] and a bottom right corner at land[0][0]. The second group has a top left corner at land[1][1] and a bottom right corner at land[2][2].

Example 2:

Input: land = [[1,1],[1,1]] Output: [[0,0,1,1]] Explanation: The first group has a top left corner at land[0][0] and a bottom right corner at land[1][1].

Example 3:

Input: land = [[0]] Output: [] Explanation: There are no groups of farmland.

Constraints:

m == land.length n == land[i].length 1 <= m, n <= 300 land consists of only 0's and 1's. Groups of farmland are rectangular in shape.

==
Problem Number: 1351 URL: https://leetcode.com/problems/operations-on-

tree Title: 1993. Operations on Tree Problem Description: You are given a tree with n nodes numbered from 0 to n - 1 in the form of a parent array parent where parent[i] is the parent of the ith node. The root of the tree is node 0, so parent[0] = -1 since it has no parent. You want to design a data structure that allows users to lock, unlock, and upgrade nodes in the tree. The data structure should support the following functions:

Lock: Locks the given node for the given user and prevents other users from locking the same node. You may only lock a node using this function if the node is unlocked. Unlock: Unlocks the given node for the given user. You may only unlock a node using this function if it is currently locked by the same user. Upgrade: Locks the given node for the given user and unlocks all of its descendants regardless of who locked it. You may only upgrade a node if all 3 conditions are true:

The node is unlocked, It has at least one locked descendant (by any user), and It does not have any locked ancestors.

Implement the LockingTree class:

LockingTree(int[] parent) initializes the data structure with the parent array. lock(int num, int user) returns true if it is possible for the user with id user to lock the node num, or false otherwise. If it is possible, the node num will become locked by the user with id user. unlock(int num, int user) returns true if it is possible for the user with id user to unlock the node num, or false otherwise. If it is possible, the node num will become unlocked. upgrade(int num, int user) returns true if it is possible for the user with id user to upgrade the node num, or false otherwise. If it is possible, the node num will be upgraded.

Example 1:

Input ["LockingTree", "lock", "unlock", "unlock", "lock", "upgrade", "lock"] [[[-1, 0, 0, 1, 1, 2, 2]], [2, 2], [2, 3], [2, 2], [4, 5], [0, 1], [0, 1]] Output [null, true, false, true, true, true, false]

Explanation LockingTree lockingTree = new LockingTree([-1, 0, 0, 1, 1, 2, 2]); lockingTree.lock(2, 2); // return true because node 2 is unlocked. // Node 2 will now be locked by user 2. lockingTree.unlock(2, 3); // return false because user 3 cannot unlock a node locked by user 2. lockingTree.unlock(2, 2); // return true because node 2 was previously locked by user 2. // Node 2 will now be unlocked. lockingTree.lock(4, 5); // return true because node 4 is unlocked. // Node 4 will now be locked by user 5. lockingTree.upgrade(0, 1); // return true because node 0 is unlocked and has at least one locked descendant (node 4). // Node 0 will now be locked by user 1 and node 4 will now be unlocked. lockingTree.lock(0, 1); // return false because node 0 is already locked.

Constraints:

n == parent.length 2 <= n <= 2000 0 <= parent[i] <= n - 1 for i != 0 parent[0] == -1 0 <= num <= n - 1 1 <= user <= 104 parent represents a valid tree.

At most 2000 calls in total will be made to lock, unlock, and upgrade.

== Problem Number: 1352 URL: https://leetcode.com/problems/the-number-of-weak-characters-in-the-game Title: 1996. The Number of Weak Characters in the Game Problem Description: You are playing a game that contains multiple characters, and each of the characters has two main properties: attack and defense. You are given a 2D integer array properties where properties[i] = [attacki, defensei] represents the properties of the ith character in the game. A character is said to be weak if any other character has both attack and defense levels strictly greater than this character's attack and defense levels. More formally, a character i is said to be weak if there exists another character j where attackj > attacki and defensej > defensei. Return the number of weak characters. Example 1: Input: properties = [[5,5],[6,3],[3,6]] Output: 0 Explanation: No character has strictly greater attack and defense than the other.

Example 2: Input: properties = [[2,2],[3,3]] Output: 1 Explanation: The first character is weak because the second character has a strictly greater attack and defense.

Example 3: Input: properties = [[1,5],[10,4],[4,3]] Output: 1 Explanation: The third character is weak because the second character has a strictly greater attack and defense.

Constraints:

2 <= properties.length <= 105 properties[i].length == 2 1 <= attacki, defensei <= 105

== Problem Number: 1353 URL: https://leetcode.com/problems/first-day-where-you-have-been-in-all-the-rooms Title: 1997. First Day Where You Have Been in All the Rooms Problem Description: There are n rooms you need to visit, labeled from 0 to n - 1. Each day is labeled, starting from 0. You will go in and visit one room a day. Initially on day 0, you visit room 0. The order you visit the rooms for the coming days is determined by the following rules and a given 0-indexed array nextVisit of length n:

Assuming that on a day, you visit room i, if you have been in room i an odd number of times (including the current visit), on the next day you will visit a room with a lower or equal room number specified by nextVisit[i] where 0 <= nextVisit[i] <= i; if you have been in room i an even number of times (including the current visit), on the next day you will visit room (i + 1) mod n.

Return the label of the first day where you have been in all the rooms. It can be shown that such a day exists. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: nextVisit = [0,0] Output: 2 Explanation: - On day 0, you visit room 0. The total times you have been in room 0 is 1, which is odd. On the next day you will visit room nextVisit[0] = 0 - On day

1, you visit room 0, The total times you have been in room 0 is 2, which is even. On the next day you will visit room (0 + 1) mod 2 = 1 - On day 2, you visit room 1. This is the first day where you have been in all the rooms.

Example 2: Input: nextVisit = [0,0,2] Output: 6 Explanation: Your room visiting order for each day is: [0,0,1,0,0,1,2,...]. Day 6 is the first day where you have been in all the rooms.

Example 3: Input: nextVisit = [0,1,2,0] Output: 6 Explanation: Your room visiting order for each day is: [0,0,1,1,2,2,3,...]. Day 6 is the first day where you have been in all the rooms.

Constraints:

n == nextVisit.length 2 <= n <= 105 0 <= nextVisit[i] <= i

==

Problem Number: 1354 URL: https://leetcode.com/problems/number-of-pairs-of-interchangeable-rectangles Title: 2001. Number of Pairs of Interchangeable Rectangles Problem Description: You are given n rectangles represented by a 0-indexed 2D integer array rectangles, where rectangles[i] = [widthi, heighti] denotes the width and height of the ith rectangle. Two rectangles i and j (i < j) are considered interchangeable if they have the same width-to-height ratio. More formally, two rectangles are interchangeable if widthi/heighti == widthj/heightj (using decimal division, not integer division). Return the number of pairs of interchangeable rectangles in rectangles. Example 1: Input: rectangles = [[4,8],[3,6],[10,20],[15,30]] Output: 6 Explanation: The following are the interchangeable pairs of rectangles by index (0-indexed): - Rectangle 0 with rectangle 1: 4/8 == 3/6. - Rectangle 0 with rectangle 2: 4/8 == 10/20. - Rectangle 0 with rectangle 3: 4/8 == 15/30. - Rectangle 1 with rectangle 2: 3/6 == 10/20. - Rectangle 1 with rectangle 3: 3/6 == 15/30. - Rectangle 2 with rectangle 3: 10/20 == 15/30.

Example 2: Input: rectangles = [[4,5],[7,8]] Output: 0 Explanation: There are no interchangeable pairs of rectangles.

Constraints:

n == rectangles.length 1 <= n <= 105 rectangles[i].length == 2 1 <= widthi, heighti <= 105

==

Problem Number: 1355 URL: https://leetcode.com/problems/maximum-product-of-the-length-of-two-palindromic-subsequences Title: 2002. Maximum Product of the Length of Two Palindromic Subsequences Problem Description: Given a string s, find two disjoint palindromic subsequences of s such that the product of their lengths is maximized. The two subsequences are disjoint if they do not both pick a character at the same index. Return the maximum possible product of the lengths of the two palindromic subsequences. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. A string is palindromic if it reads the same forward and backward.   Example 1:

Input: s = "leetcodecom" Output: 9 Explanation: An optimal solution is to choose "ete" for the 1st subsequence and "cdc" for the 2nd subsequence. The product of their lengths is: 3 * 3 = 9.

Example 2: Input: s = "bb" Output: 1 Explanation: An optimal solution is to choose "b" (the first character) for the 1st subsequence and "b" (the second character) for the 2nd subsequence. The product of their lengths is: 1 * 1 = 1.

Example 3: Input: s = "accbcaxxcxx" Output: 25 Explanation: An optimal solution is to choose "accca" for the 1st subsequence and "xxcxx" for the 2nd subsequence. The product of their lengths is: 5 * 5 = 25.

Constraints:

2 <= s.length <= 12 s consists of lowercase English letters only.

==
Problem Number: 1356 URL: https://leetcode.com/problems/find-original-array-from-doubled-array Title: 2007. Find Original Array From Doubled Array Problem Description: An integer array original is transformed into a doubled array changed by appending twice the value of every element in original, and then randomly shuffling the resulting array. Given an array changed, return original if changed is a doubled array. If changed is not a doubled array, return an empty array. The elements in original may be returned in any order.   Example 1: Input: changed = [1,3,4,2,6,8] Output: [1,3,4] Explanation: One possible original array could be [1,3,4]: - Twice the value of 1 is 1 * 2 = 2. - Twice the value of 3 is 3 * 2 = 6. - Twice the value of 4 is 4 * 2 = 8. Other original arrays could be [4,3,1] or [3,1,4].

Example 2: Input: changed = [6,3,0,1] Output: [] Explanation: changed is not a doubled array.

Example 3: Input: changed = [1] Output: [] Explanation: changed is not a doubled array.

Constraints:

1 <= changed.length <= 105 0 <= changed[i] <= 105

==
Problem Number: 1357 URL: https://leetcode.com/problems/maximum-earnings-from-taxi Title: 2008. Maximum Earnings From Taxi Problem Description: There are n points on a road you are driving your taxi on. The n points on the road are labeled from 1 to n in the direction you are going, and you want to drive from point 1 to point n to make money by picking up passengers. You cannot change the direction of the taxi. The passengers are represented by a 0-indexed 2D integer array rides, where rides[i] = [starti, endi, tipi] denotes the ith passenger requesting a ride from point starti to point endi

who is willing to give a tipi dollar tip. For each passenger i you pick up, you earn endi - starti + tipi dollars. You may only drive at most one passenger at a time. Given n and rides, return the maximum number of dollars you can earn by picking up the passengers optimally. Note: You may drop off a passenger and pick up a different passenger at the same point. Example 1: Input: n = 5, rides = [[2,5,4],[1,5,1]] Output: 7 Explanation: We can pick up passenger 0 to earn 5 - 2 + 4 = 7 dollars.

Example 2: Input: n = 20, rides = [[1,6,1],[3,10,2],[10,12,3],[11,12,2],[12,15,2],[13,18,1]] Output: 20 Explanation: We will pick up the following passengers: - Drive passenger 1 from point 3 to point 10 for a profit of 10 - 3 + 2 = 9 dollars. - Drive passenger 2 from point 10 to point 12 for a profit of 12 - 10 + 3 = 5 dollars. - Drive passenger 5 from point 13 to point 18 for a profit of 18 - 13 + 1 = 6 dollars. We earn 9 + 5 + 6 = 20 dollars in total. Constraints:

1 <= n <= 105 1 <= rides.length <= 3 * 104 rides[i].length == 3 1 <= starti < endi <= n 1 <= tipi <= 105

==

Problem Number: 1358 URL: https://leetcode.com/problems/sum-of-beauty-in-the-array Title: 2012. Sum of Beauty in the Array Problem Description: You are given a 0-indexed integer array nums. For each index i (1 <= i <= nums.length - 2) the beauty of nums[i] equals:

2, if nums[j] < nums[i] < nums[k], for all 0 <= j < i and for all i < k <= nums.length - 1. 1, if nums[i - 1] < nums[i] < nums[i + 1], and the previous condition is not satisfied. 0, if none of the previous conditions holds.

Return the sum of beauty of all nums[i] where 1 <= i <= nums.length - 2. Example 1: Input: nums = [1,2,3] Output: 2 Explanation: For each index i in the range 1 <= i <= 1: - The beauty of nums[1] equals 2.

Example 2: Input: nums = [2,4,6,4] Output: 1 Explanation: For each index i in the range 1 <= i <= 2: - The beauty of nums[1] equals 1. - The beauty of nums[2] equals 0.

Example 3: Input: nums = [3,2,1] Output: 0 Explanation: For each index i in the range 1 <= i <= 1: - The beauty of nums[1] equals 0.

Constraints:

3 <= nums.length <= 105 1 <= nums[i] <= 105

==

Problem Number: 1359 URL: https://leetcode.com/problems/detect-squares Title: 2013. Detect Squares Problem Description: You are given a stream of points on the X-Y plane. Design an algorithm that:

Adds new points from the stream into a data structure. Duplicate points are allowed and should be treated as different points. Given a query point, counts the number of ways to choose three points from the data structure such that the three points and the query point form an axis-aligned square with positive area.

An axis-aligned square is a square whose edges are all the same length and are either parallel or perpendicular to the x-axis and y-axis. Implement the DetectSquares class:

DetectSquares() Initializes the object with an empty data structure. void add(int[] point) Adds a new point point = [x, y] to the data structure. int count(int[] point) Counts the number of ways to form axis-aligned squares with point point = [x, y] as described above.

Example 1:

Input ["DetectSquares", "add", "add", "add", "count", "count", "add", "count"] [[], [[3, 10]], [[11, 2]], [[3, 2]], [[11, 10]], [[14, 8]], [[11, 2]], [[11, 10]]] Output [null, null, null, null, 1, 0, null, 2]

Explanation DetectSquares detectSquares = new DetectSquares(); detectSquares.add([3, 10]); detectSquares.add([11, 2]); detectSquares.add([3, 2]); detectSquares.count([11, 10]); // return 1. You can choose: // - The first, second, and third points detectSquares.count([14, 8]); // return 0. The query point cannot form a square with any points in the data structure. detectSquares.add([11, 2]); // Adding duplicate points is allowed. detectSquares.count([11, 10]); // return 2. You can choose: // - The first, second, and third points // - The first, third, and fourth points

Constraints:

point.length == 2 0 <= x, y <= 1000 At most 3000 calls in total will be made to add and count.

==
Problem Number: 1360 URL: https://leetcode.com/problems/grid-game Title: 2017. Grid Game Problem Description: You are given a 0-indexed 2D array grid of size 2 x n, where grid[r][c] represents the number of points at position (r, c) on the matrix. Two robots are playing a game on this matrix. Both robots initially start at (0, 0) and want to reach (1, n-1). Each robot may only move to the right ((r, c) to (r, c + 1)) or down ((r, c) to (r + 1, c)). At the start of the game, the first robot moves from (0, 0) to (1, n-1), collecting all the points from the cells on its path. For all cells (r, c) traversed on the path, grid[r][c] is set to 0. Then, the second robot moves from (0, 0) to (1, n-1), collecting the points on its path. Note that their paths may intersect with one another. The first robot wants to minimize the number of points collected by the second robot. In contrast, the second robot wants to maximize the number of points it collects. If both robots play optimally, return the number of points collected by the second robot. Example 1:

Input: grid = [[2,5,4],[1,5,1]] Output: 4 Explanation: The optimal path taken by the first robot is shown in red, and the optimal path taken by the second robot is shown in blue. The cells visited by the first robot are set to 0. The second robot will collect 0 + 0 + 4 + 0 = 4 points.

Example 2:

Input: grid = [[3,3,1],[8,5,2]] Output: 4 Explanation: The optimal path taken by the first robot is shown in red, and the optimal path taken by the second robot is shown in blue. The cells visited by the first robot are set to 0. The second robot will collect 0 + 3 + 1 + 0 = 4 points.

Example 3:

Input: grid = [[1,3,1,15],[1,3,3,1]] Output: 7 Explanation: The optimal path taken by the first robot is shown in red, and the optimal path taken by the second robot is shown in blue. The cells visited by the first robot are set to 0. The second robot will collect 0 + 1 + 3 + 3 + 0 = 7 points.

Constraints:

grid.length == 2 n == grid[r].length 1 <= n <= 5 * 104 1 <= grid[r][c] <= 105

==
Problem Number: 1361 URL: https://leetcode.com/problems/check-if-word-can-be-placed-in-crossword Title: 2018. Check if Word Can Be Placed In Crossword Problem Description: You are given an m x n matrix board, representing the current state of a crossword puzzle. The crossword contains lowercase English letters (from solved words), ' ' to represent any empty cells, and '#' to represent any blocked cells. A word can be placed horizontally (left to right or right to left) or vertically (top to bottom or bottom to top) in the board if:

It does not occupy a cell containing the character '#'. The cell each letter is placed in must either be ' ' (empty) or match the letter already on the board. There must not be any empty cells ' ' or other lowercase letters directly left or right of the word if the word was placed horizontally. There must not be any empty cells ' ' or other lowercase letters directly above or below the word if the word was placed vertically.

Given a string word, return true if word can be placed in board, or false otherwise. Example 1:

Input: board = [["#", " ", "#"], [" ", " ", "#"], ["#", "c", " "]], word = "abc" Output: true Explanation: The word "abc" can be placed as shown above (top to bottom).

Example 2:

Input: board = [[" ", "#", "a"], [" ", "#", "c"], [" ", "#", "a"]], word = "ac" Output: false Explanation: It is impossible to place the word because there will always be a space/letter above or below it. Example 3:

Input: board = [["#", " ", "#"], [" ", " ", "#"], ["#", " ", "c"]], word = "ca" Output: true Explanation: The word "ca" can be placed as shown above (right to left).

Constraints:

m == board.length n == board[i].length 1 <= m * n <= 2 * 105 board[i][j] will be ' ', '#', or a lowercase English letter. 1 <= word.length <= max(m, n) word will contain only lowercase English letters.

==
Problem Number: 1362 URL: https://leetcode.com/problems/number-of-pairs-of-strings-with-concatenation-equal-to-target Title: 2023. Number of Pairs of Strings With Concatenation Equal to Target Problem Description: Given an array of digit strings nums and a digit string target, return the number of pairs of indices (i, j) (where i != j) such that the concatenation of nums[i] + nums[j] equals target. Example 1: Input: nums = ["777","7","77","77"], target = "7777" Output: 4 Explanation: Valid pairs are: - (0, 1): "777" + "7" - (1, 0): "7" + "777" - (2, 3): "77" + "77" - (3, 2): "77" + "77"

Example 2: Input: nums = ["123","4","12","34"], target = "1234" Output: 2 Explanation: Valid pairs are: - (0, 1): "123" + "4" - (2, 3): "12" + "34"

Example 3: Input: nums = ["1","1","1"], target = "11" Output: 6 Explanation: Valid pairs are: - (0, 1): "1" + "1" - (1, 0): "1" + "1" - (0, 2): "1" + "1" - (2, 0): "1" + "1" - (1, 2): "1" + "1" - (2, 1): "1" + "1"

Constraints:

2 <= nums.length <= 100 1 <= nums[i].length <= 100 2 <= target.length <= 100 nums[i] and target consist of digits. nums[i] and target do not have leading zeros.

==
Problem Number: 1363 URL: https://leetcode.com/problems/maximize-the-confusion-of-an-exam Title: 2024. Maximize the Confusion of an Exam Problem Description: A teacher is writing a test with n true/false questions, with 'T' denoting true and 'F' denoting false. He wants to confuse the students by maximizing the number of consecutive questions with the same answer (multiple trues or multiple falses in a row). You are given a string answerKey, where answerKey[i] is the original answer to the ith question. In addition, you are given an integer k, the maximum number of times you may perform the following operation:

Change the answer key for any question to 'T' or 'F' (i.e., set answerKey[i] to 'T' or 'F').

Return the maximum number of consecutive 'T's or 'F's in the answer key after performing the operation at most k times. Example 1: Input: answerKey = "TTFF", k = 2 Output: 4 Explanation: We can replace both the 'F's with 'T's to make answerKey = "TTTT". There are four consecutive 'T's.

Example 2: Input: answerKey = "TFFT", k = 1 Output: 3 Explanation: We can replace the first 'T' with an 'F' to make answerKey = "FFFT". Alternatively, we can replace the second 'T' with an 'F' to make answerKey = "TFFF". In both cases, there are three consecutive 'F's.

Example 3: Input: answerKey = "TTFTTFTT", k = 1 Output: 5 Explanation: We can replace the first 'F' to make answerKey = "TTTTTFTT" Alternatively, we can replace the second 'F' to make answerKey = "TTFTTTTT". In both cases, there are five consecutive 'T's.

Constraints:

n == answerKey.length 1 <= n <= 5 * 104 answerKey[i] is either 'T' or 'F' 1 <= k <= n

==

Problem Number: 1364 URL: https://leetcode.com/problems/find-missing-observations Title: 2028. Find Missing Observations Problem Description: You have observations of n + m 6-sided dice rolls with each face numbered from 1 to 6. n of the observations went missing, and you only have the observations of m rolls. Fortunately, you have also calculated the average value of the n + m rolls. You are given an integer array rolls of length m where rolls[i] is the value of the ith observation. You are also given the two integers mean and n. Return an array of length n containing the missing observations such that the average value of the n + m rolls is exactly mean. If there are multiple valid answers, return any of them. If no such array exists, return an empty array. The average value of a set of k numbers is the sum of the numbers divided by k. Note that mean is an integer, so the sum of the n + m rolls should be divisible by n + m.   Example 1: Input: rolls = [3,2,4,3], mean = 4, n = 2 Output: [6,6] Explanation: The mean of all n + m rolls is (3 + 2 + 4 + 3 + 6 + 6) / 6 = 4.

Example 2: Input: rolls = [1,5,6], mean = 3, n = 4 Output: [2,3,2,2] Explanation: The mean of all n + m rolls is (1 + 5 + 6 + 2 + 3 + 2 + 2) / 7 = 3.

Example 3: Input: rolls = [1,2,3,4], mean = 6, n = 4 Output: [] Explanation: It is impossible for the mean to be 6 no matter what the 4 missing rolls are.

Constraints:

m == rolls.length 1 <= n, m <= 105 1 <= rolls[i], mean <= 6

==

Problem Number: 1365 URL: https://leetcode.com/problems/stone-game-ix Title: 2029. Stone Game IX Problem Description: Alice and Bob continue their games with stones. There is a row of n stones, and each stone has an associated value. You are given an integer array stones, where stones[i] is the value of the ith stone. Alice and Bob take turns, with Alice starting first. On each turn, the player may remove any stone from stones. The player who removes a stone loses if the sum of the values of all removed stones is divisible by 3. Bob will win automatically if there are no remaining stones (even if it is Alice's turn). Assuming both players play optimally, return true if Alice wins and false if Bob wins.   Example 1: Input: stones = [2,1] Output: true Explanation: The game will be played as follows: - Turn 1: Alice can remove either stone. - Turn 2: Bob removes the remaining stone. The sum of the removed stones is 1 + 2 = 3 and is divisible by 3. Therefore, Bob loses and Alice wins the game.

Example 2: Input: stones = [2] Output: false Explanation: Alice will remove the only stone, and the sum of the values on the removed stones is 2. Since all the stones are removed and the sum of values is not divisible by 3, Bob wins the game.

Example 3: Input: stones = [5,1,2,4,3] Output: false Explanation: Bob will always win. One possible way for Bob to win is shown below: - Turn 1: Alice can remove the second stone with value 1. Sum of removed stones = 1. - Turn 2: Bob removes the fifth stone with value 3. Sum of removed stones = 1 + 3 = 4. - Turn 3: Alices removes the fourth stone with value 4. Sum of removed stones = 1 + 3 + 4 = 8. - Turn 4: Bob removes the third stone with value 2. Sum of removed stones = 1 + 3 + 4 + 2 = 10. - Turn 5: Alice removes the first stone with value 5. Sum of removed stones = 1 + 3 + 4 + 2 + 5 = 15. Alice loses the game because the sum of the removed stones (15) is divisible by 3. Bob wins the game.

Constraints:

1 <= stones.length <= 105 1 <= stones[i] <= 104

==

Problem Number: 1366 URL: https://leetcode.com/problems/minimum-operations-to-make-a-uni-value-grid Title: 2033. Minimum Operations to Make a Uni-Value Grid Problem Description: You are given a 2D integer grid of size m x n and an integer x. In one operation, you can add x to or subtract x from any element in the grid. A uni-value grid is a grid where all the elements of it are equal. Return the minimum number of operations to make the grid uni-value. If it is not possible, return -1.   Example 1:

Input: grid = [[2,4],[6,8]], x = 2 Output: 4 Explanation: We can make every element equal to 4 by doing the following: - Add x to 2 once. - Subtract x from 6 once. - Subtract x from 8 twice. A total of 4 operations were used.

Example 2:

Input: grid = [[1,5],[2,3]], x = 1 Output: 5 Explanation: We can make every element equal to 3.

Example 3:

Input: grid = [[1,2],[3,4]], x = 2 Output: -1 Explanation: It is impossible to make every element equal.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 1 <= m * n <= 105 1 <= x, grid[i][j] <= 104

==

Problem Number: 1367 URL: https://leetcode.com/problems/stock-price-fluctuation Title: 2034. Stock Price Fluctuation Problem Description: You are given a stream of records about a particular stock. Each record contains a timestamp and the corresponding price of the stock at that timestamp. Unfortunately due to the volatile nature of the stock market, the records do not come in order. Even worse, some records may be incorrect. Another record with the same timestamp may appear later in the stream correcting the price of the previous wrong record. Design an algorithm that:

Updates the price of the stock at a particular timestamp, correcting the price from any previous records at the timestamp. Finds the latest price of the stock based on the current records. The latest price is the price at the latest timestamp recorded. Finds the maximum price the stock has been based on the current records. Finds the minimum price the stock has been based on the current records.

Implement the StockPrice class:

StockPrice() Initializes the object with no price records. void update(int timestamp, int price) Updates the price of the stock at the given timestamp. int current() Returns the latest price of the stock. int maximum() Returns the maximum price of the stock. int minimum() Returns the minimum price of the stock.

Example 1: Input ["StockPrice", "update", "update", "current", "maximum", "update", "maximum", "update", "minimum"] [[], [1, 10], [2, 5], [], [], [1, 3], [], [4, 2], []] Output [null, null, null, 5, 10, null, 5, null, 2]

Explanation StockPrice stockPrice = new StockPrice(); stockPrice.update(1, 10); // Timestamps are [1] with corresponding prices [10]. stockPrice.update(2, 5); // Timestamps are [1,2] with corresponding prices [10,5]. stockPrice.current(); // return 5, the latest timestamp is 2 with the price being 5. stockPrice.maximum(); // return 10, the maximum price is 10 at timestamp 1. stockPrice.update(1, 3); // The previous timestamp 1 had the wrong price, so it is updated to 3. // Timestamps are [1,2] with corresponding prices [3,5]. stockPrice.maximum(); // return 5, the maximum price is 5 after the correction. stockPrice.update(4, 2); // Timestamps are [1,2,4] with corresponding prices [3,5,2]. stockPrice.minimum(); // return 2, the minimum price is 2 at timestamp 4.

Constraints:

1 <= timestamp, price <= 109 At most 105 calls will be made in total to update, current, maximum, and minimum. current, maximum, and minimum will be called only after update has been called at least once.

==
Problem Number: 1368 URL: https://leetcode.com/problems/remove-colored-pieces-if-both-neighbors-are-the-same-color Title: 2038. Remove Colored Pieces if Both Neighbors are the Same Color Problem Description: There are n pieces arranged in a line, and each piece is colored either by 'A' or by 'B'. You are given a string colors of length n where colors[i] is the color of the ith piece. Alice and Bob are playing a game where they take alternating turns removing pieces from the line. In this game, Alice moves first.

Alice is only allowed to remove a piece colored 'A' if both its neighbors are also colored 'A'. She is not allowed to remove pieces that are colored 'B'. Bob is only allowed to remove a piece colored 'B' if both its neighbors are also colored 'B'. He is not allowed to remove pieces that are colored 'A'. Alice and Bob cannot remove pieces from the edge of the line. If a player cannot make a move on their turn, that player loses and the other player wins.

Assuming Alice and Bob play optimally, return true if Alice wins, or return false if Bob wins. Example 1: Input: colors = "AAABABB" Output: true Explanation: AAABABB -> AABABB Alice moves first. She removes the second 'A' from the left since that is the only 'A' whose neighbors are both 'A'.

Now it's Bob's turn. Bob cannot make a move on his turn since there are no 'B's whose neighbors are both 'B'. Thus, Alice wins, so return true.

Example 2: Input: colors = "AA" Output: false Explanation: Alice has her turn first. There are only two 'A's and both are on the edge of the line, so she cannot move on her turn. Thus, Bob wins, so return false.

Example 3: Input: colors = "ABBBBBBBAAA" Output: false Explanation: ABBBBBBBAAA -> ABBBBBBBAA Alice moves first. Her only option is to remove the second to last 'A' from the right.

ABBBBBBBAA -> ABBBBBBAA Next is Bob's turn. He has many options for which 'B' piece to remove. He can pick any.

On Alice's second turn, she has no more pieces that she can remove. Thus, Bob wins, so return false.

Constraints:

1 <= colors.length <= 105 colors consists of only the letters 'A' and 'B'

==
Problem Number: 1369 URL: https://leetcode.com/problems/the-time-when-the-network-becomes-idle Title: 2039. The Time When the Network Becomes Idle Problem Description: There is a network of n servers, labeled from 0 to n - 1. You are given a 2D integer array edges, where edges[i] = [ui, vi] indicates there is a message channel between servers ui and vi, and they can pass any number of messages to each other directly in one second. You are also given a 0-indexed integer array patience of length n. All servers are connected, i.e., a message can be passed from one server to any other server(s) directly or

indirectly through the message channels. The server labeled 0 is the master server. The rest are data servers. Each data server needs to send its message to the master server for processing and wait for a reply. Messages move between servers optimally, so every message takes the least amount of time to arrive at the master server. The master server will process all newly arrived messages instantly and send a reply to the originating server via the reversed path the message had gone through. At the beginning of second 0, each data server sends its message to be processed. Starting from second 1, at the beginning of every second, each data server will check if it has received a reply to the message it sent (including any newly arrived replies) from the master server:

If it has not, it will resend the message periodically. The data server i will resend the message every patience[i] second(s), i.e., the data server i will resend the message if patience[i] second(s) have elapsed since the last time the message was sent from this server. Otherwise, no more resending will occur from this server.

The network becomes idle when there are no messages passing between servers or arriving at servers. Return the earliest second starting from which the network becomes idle. Example 1:

Input: edges = [[0,1],[1,2]], patience = [0,2,1] Output: 8 Explanation: At (the beginning of) second 0, - Data server 1 sends its message (denoted 1A) to the master server. - Data server 2 sends its message (denoted 2A) to the master server.

At second 1, - Message 1A arrives at the master server. Master server processes message 1A instantly and sends a reply 1A back. - Server 1 has not received any reply. 1 second (1 < patience[1] = 2) elapsed since this server has sent the message, therefore it does not resend the message. - Server 2 has not received any reply. 1 second (1 == patience[2] = 1) elapsed since this server has sent the message, therefore it resends the message (denoted 2B).

At second 2, - The reply 1A arrives at server 1. No more resending will occur from server 1. - Message 2A arrives at the master server. Master server processes message 2A instantly and sends a reply 2A back. - Server 2 resends the message (denoted 2C). ... At second 4, - The reply 2A arrives at server 2. No more resending will occur from server 2. ... At second 7, reply 2D arrives at server 2.

Starting from the beginning of the second 8, there are no messages passing between servers or arriving at servers. This is the time when the network becomes idle.

Example 2:

Input: edges = [[0,1],[0,2],[1,2]], patience = [0,10,10] Output: 3 Explanation: Data servers 1 and 2 receive a reply back at the beginning of second 2. From the beginning of the second 3, the network becomes idle.

Constraints:

n == patience.length 2 <= n <= 105 patience[0] == 0 1 <= patience[i] <= 105 for 1 <= i < n 1 <= edges.length <= min(105, n * (n - 1) / 2) edges[i].length == 2 0 <= ui, vi < n ui != vi There are no duplicate edges. Each server can directly or indirectly reach another server.

==
Problem Number: 1370 URL: https://leetcode.com/problems/simple-bank-system Title: 2043. Simple Bank System Problem Description: You have been tasked with writing a program for a popular bank that will automate all its incoming transactions (transfer, deposit, and withdraw). The bank has n accounts numbered from 1 to n. The initial balance of each account is stored in a 0-indexed integer array balance, with the (i + 1)th account having an initial balance of balance[i]. Execute all the valid transactions. A transaction is valid if:

The given account number(s) are between 1 and n, and The amount of money withdrawn or transferred from is less than or equal to the balance of the account.

Implement the Bank class:

Bank(long[] balance) Initializes the object with the 0-indexed integer array balance. boolean transfer(int account1, int account2, long money) Transfers money dollars from the account numbered account1 to the account numbered account2. Return true if the transaction was successful, false otherwise. boolean deposit(int account, long money) Deposit money dollars into the account numbered account. Return true if the transaction was successful, false otherwise. boolean withdraw(int account, long money) Withdraw money dollars from the account numbered account. Return true if the transaction was successful, false otherwise.

Example 1: Input ["Bank", "withdraw", "transfer", "deposit", "transfer", "withdraw"] [[[10, 100, 20, 50, 30]], [3, 10], [5, 1, 20], [5, 20], [3, 4, 15], [10, 50]] Output [null, true, true, true, false, false]

Explanation Bank bank = new Bank([10, 100, 20, 50, 30]); bank.withdraw(3, 10); // return true, account 3 has a balance of $20, so it is valid to withdraw $10. // Account 3 has $20 - $10 = $10. bank.transfer(5, 1, 20); // return true, account 5 has a balance of $30, so it is valid to transfer $20. // Account 5 has $30 - $20 = $10, and account 1 has $10 + $20 = $30. bank.deposit(5, 20); // return true, it is valid to deposit $20 to account 5. // Account 5 has $10 + $20 = $30. bank.transfer(3, 4, 15); // return false, the current balance of account 3 is $10, // so it is invalid to transfer $15 from it. bank.withdraw(10, 50); // return false, it is invalid because account 10 does not exist.

Constraints:

n == balance.length 1 <= n, account, account1, account2 <= 105 0 <= balance[i], money <= 1012 At most 104 calls will be made to each function transfer, deposit, withdraw.

==
Problem Number: 1371 URL: https://leetcode.com/problems/count-number-of-maximum-bitwise-or-subsets Title: 2044. Count Number of Maximum Bitwise-OR Subsets Problem Description: Given an integer array nums, find the maximum possible bitwise OR of a subset of nums and return the number of different non-empty subsets with the maximum bitwise OR. An array a is a subset of an array b if a can be obtained from b by deleting some (possibly zero) elements of b. Two subsets are considered different if the indices of the elements chosen are different. The bitwise OR of an array a is equal to a[0] OR a[1] OR ... OR a[a.length - 1] (0-indexed). Example 1: Input: nums = [3,1] Output: 2 Explanation: The maximum possible bitwise OR of a subset is 3. There are 2 subsets with a bitwise OR of 3: - [3] - [3,1]

Example 2: Input: nums = [2,2,2] Output: 7 Explanation: All non-empty subsets of [2,2,2] have a bitwise OR of 2. There are 23 - 1 = 7 total subsets.

Example 3: Input: nums = [3,2,1,5] Output: 6 Explanation: The maximum possible bitwise OR of a subset is 7. There are 6 subsets with a bitwise OR of 7: - [3,5] - [3,1,5] - [3,2,5] - [3,2,1,5] - [2,5] - [2,1,5] Constraints:

1 <= nums.length <= 16 1 <= nums[i] <= 105

==
Problem Number: 1372 URL: https://leetcode.com/problems/next-greater-numerically-balanced-number Title: 2048. Next Greater Numerically Balanced Number Problem Description: An integer x is numerically balanced if for every digit d in the number x, there are exactly d occurrences of that digit in x. Given an integer n, return the smallest numerically balanced number strictly greater than n. Example 1: Input: n = 1 Output: 22 Explanation: 22 is numerically balanced since: - The digit 2 occurs 2 times. It is also the smallest numerically balanced number strictly greater than 1.

Example 2: Input: n = 1000 Output: 1333 Explanation: 1333 is numerically balanced since: - The digit 1 occurs 1 time. - The digit 3 occurs 3 times. It is also the smallest numerically balanced number strictly greater than 1000. Note that 1022 cannot be the answer because 0 appeared more than 0 times.

Example 3: Input: n = 3000 Output: 3133 Explanation: 3133 is numerically balanced since: - The digit 1 occurs 1 time. - The digit 3 occurs 3 times. It is also the smallest numerically balanced number strictly greater than 3000.

Constraints:

0 <= n <= 106

==
Problem Number: 1373 URL: https://leetcode.com/problems/count-nodes-with-the-highest-score Title: 2049. Count Nodes With the Highest Score Problem Description: There is a binary tree rooted at 0 consisting of n nodes. The nodes are labeled from 0 to n - 1. You are given a 0-indexed integer array

parents representing the tree, where parents[i] is the parent of node i. Since node 0 is the root, parents[0] == -1. Each node has a score. To find the score of a node, consider if the node and the edges connected to it were removed. The tree would become one or more non-empty subtrees. The size of a subtree is the number of the nodes in it. The score of the node is the product of the sizes of all those subtrees. Return the number of nodes that have the highest score.   Example 1:

Input: parents = [-1,2,0,2,0] Output: 3 Explanation: - The score of node 0 is: 3 * 1 = 3 - The score of node 1 is: 4 = 4 - The score of node 2 is: 1 * 1 * 2 = 2 - The score of node 3 is: 4 = 4 - The score of node 4 is: 4 = 4 The highest score is 4, and three nodes (node 1, node 3, and node 4) have the highest score.

Example 2:

Input: parents = [-1,2,0] Output: 2 Explanation: - The score of node 0 is: 2 = 2 - The score of node 1 is: 2 = 2 - The score of node 2 is: 1 * 1 = 1 The highest score is 2, and two nodes (node 0 and node 1) have the highest score.

Constraints:

n == parents.length 2 <= n <= 105 parents[0] == -1 0 <= parents[i] <= n - 1 for i != 0 parents represents a valid binary tree.

==

Problem Number: 1374 URL: https://leetcode.com/problems/two-best-non-overlapping-events Title: 2054. Two Best Non-Overlapping Events Problem Description: You are given a 0-indexed 2D integer array of events where events[i] = [startTimei, endTimei, valuei]. The ith event starts at startTimei and ends at endTimei, and if you attend this event, you will receive a value of valuei. You can choose at most two non-overlapping events to attend such that the sum of their values is maximized. Return this maximum sum. Note that the start time and end time is inclusive: that is, you cannot attend two events where one of them starts and the other ends at the same time. More specifically, if you attend an event with end time t, the next event must start at or after t + 1.   Example 1:

Input: events = [[1,3,2],[4,5,2],[2,4,3]] Output: 4 Explanation: Choose the green events, 0 and 1 for a sum of 2 + 2 = 4.

Example 2:

Input: events = [[1,3,2],[4,5,2],[1,5,5]] Output: 5 Explanation: Choose event 2 for a sum of 5.

Example 3:

Input: events = [[1,5,3],[1,5,1],[6,6,5]] Output: 8 Explanation: Choose events 0 and 2 for a sum of 3 + 5 = 8.   Constraints:

2 <= events.length <= 105 events[i].length == 3 1 <= startTimei <= endTimei <= 109 1 <= valuei <= 106

==

Problem Number: 1375 URL: https://leetcode.com/problems/plates-between-candles Title: 2055. Plates Between Candles Problem Description: There is a long table with a line of plates and candles arranged on top of it. You are given a 0-indexed string s consisting of characters ’*’ and ’|’ only, where a ’*’ represents a plate and a ’|’ represents a candle. You are also given a 0-indexed 2D integer array queries where queries[i] = [lefti, righti] denotes the substring s[lefti...righti] (inclusive). For each query, you need to find the number of plates between candles that are in the substring. A plate is considered between candles if there is at least one candle to its left and at least one candle to its right in the substring.

For example, s = ”||**||**|*”, and a query [3, 8] denotes the substring ”*||**|”. The number of plates between candles in this substring is 2, as each of the two plates has at least one candle in the substring to its left and right.

Return an integer array answer where answer[i] is the answer to the ith query. Example 1:

Input: s = ”**|**|***|”, queries = [[2,5],[5,9]] Output: [2,3] Explanation: - queries[0] has two plates between candles. - queries[1] has three plates between candles.

Example 2:

Input: s = ”***|**|*****|**||**|*”, queries = [[1,17],[4,5],[14,17],[5,11],[15,16]] Output: [9,0,0,0,0] Explanation: - queries[0] has nine plates between candles. - The other queries have zero plates between candles.

Constraints:

3 <= s.length <= 105 s consists of ’*’ and ’|’ characters. 1 <= queries.length <= 105 queries[i].length == 2 0 <= lefti <= righti < s.length

==

Problem Number: 1376 URL: https://leetcode.com/problems/find-the-minimum-and-maximum-number-of-nodes-between-critical-points Title: 2058. Find the Minimum and Maximum Number of Nodes Between Critical Points Problem Description: A critical point in a linked list is defined as either a local maxima or a local minima. A node is a local maxima if the current node has a value strictly greater than the previous node and the next node. A node is a local minima if the current node has a value strictly smaller than the previous node and the next node. Note that a node can only be a local maxima/minima if there exists both a previous node and a next node. Given a linked list head, return an array of length 2 containing [minDistance, maxDistance] where minDistance is the minimum distance between any two distinct critical points and maxDistance is the maximum distance between any two distinct critical points. If there are fewer than two critical points, return [-1, -1]. Example 1:

Input: head = [3,1] Output: [-1,-1] Explanation: There are no critical points in

[3,1].

Example 2:

Input: head = [5,3,1,2,5,1,2] Output: [1,3] Explanation: There are three critical points: - [5,3,1,2,5,1,2]: The third node is a local minima because 1 is less than 3 and 2. - [5,3,1,2,5,1,2]: The fifth node is a local maxima because 5 is greater than 2 and 1. - [5,3,1,2,5,1,2]: The sixth node is a local minima because 1 is less than 5 and 2. The minimum distance is between the fifth and the sixth node. minDistance = 6 - 5 = 1. The maximum distance is between the third and the sixth node. maxDistance = 6 - 3 = 3.

Example 3:

Input: head = [1,3,2,2,3,2,2,2,7] Output: [3,3] Explanation: There are two critical points: - [1,3,2,2,3,2,2,2,7]: The second node is a local maxima because 3 is greater than 1 and 2. - [1,3,2,2,3,2,2,2,7]: The fifth node is a local maxima because 3 is greater than 2 and 2. Both the minimum and maximum distances are between the second and the fifth node. Thus, minDistance and maxDistance is 5 - 2 = 3. Note that the last node is not considered a local maxima because it does not have a next node.

Constraints:

The number of nodes in the list is in the range [2, 105]. 1 <= Node.val <= 105

==

Problem Number: 1377 URL: https://leetcode.com/problems/minimum-operations-to-convert-number Title: 2059. Minimum Operations to Convert Number Problem Description: You are given a 0-indexed integer array nums containing distinct numbers, an integer start, and an integer goal. There is an integer x that is initially set to start, and you want to perform operations on x such that it is converted to goal. You can perform the following operation repeatedly on the number x: If 0 <= x <= 1000, then for any index i in the array (0 <= i < nums.length), you can set x to any of the following:

x + nums[i] x - nums[i] x ^ nums[i] (bitwise-XOR)

Note that you can use each nums[i] any number of times in any order. Operations that set x to be out of the range 0 <= x <= 1000 are valid, but no more operations can be done afterward. Return the minimum number of operations needed to convert x = start into goal, and -1 if it is not possible. Example 1: Input: nums = [2,4,12], start = 2, goal = 12 Output: 2 Explanation: We can go from 2 → 14 → 12 with the following 2 operations. - 2 + 12 = 14 - 14 - 2 = 12

Example 2: Input: nums = [3,5,7], start = 0, goal = -4 Output: 2 Explanation: We can go from 0 → 3 → -4 with the following 2 operations. - 0 + 3 = 3 - 3 - 7 = -4 Note that the last operation sets x out of the range 0 <= x <= 1000, which is valid.

Example 3: Input: nums = [2,8,16], start = 0, goal = 1 Output: -1 Explanation: There is no way to convert 0 into 1.

Constraints:

1 <= nums.length <= 1000 -109 <= nums[i], goal <= 109 0 <= start <= 1000 start != goal All the integers in nums are distinct.

==

Problem Number: 1378 URL: https://leetcode.com/problems/vowels-of-all-substrings Title: 2063. Vowels of All Substrings Problem Description: Given a string word, return the sum of the number of vowels ('a', 'e', 'i', 'o', and 'u') in every substring of word. A substring is a contiguous (non-empty) sequence of characters within a string. Note: Due to the large constraints, the answer may not fit in a signed 32-bit integer. Please be careful during the calculations. Example 1: Input: word = "aba" Output: 6 Explanation: All possible substrings are: "a", "ab", "aba", "b", "ba", and "a". - "b" has 0 vowels in it - "a", "ab", "ba", and "a" have 1 vowel each - "aba" has 2 vowels in it Hence, the total sum of vowels = 0 + 1 + 1 + 1 + 1 + 2 = 6.

Example 2: Input: word = "abc" Output: 3 Explanation: All possible substrings are: "a", "ab", "abc", "b", "bc", and "c". - "a", "ab", and "abc" have 1 vowel each - "b", "bc", and "c" have 0 vowels each Hence, the total sum of vowels = 1 + 1 + 1 + 0 + 0 + 0 = 3.

Example 3: Input: word = "ltcd" Output: 0 Explanation: There are no vowels in any substring of "ltcd".

Constraints:

1 <= word.length <= 105 word consists of lowercase English letters.

==

Problem Number: 1379 URL: https://leetcode.com/problems/minimized-maximum-of-products-distributed-to-any-store Title: 2064. Minimized Maximum of Products Distributed to Any Store Problem Description: You are given an integer n indicating there are n specialty retail stores. There are m product types of varying amounts, which are given as a 0-indexed integer array quantities, where quantities[i] represents the number of products of the ith product type. You need to distribute all products to the retail stores following these rules:

A store can only be given at most one product type but can be given any amount of it. After distribution, each store will have been given some number of products (possibly 0). Let x represent the maximum number of products given to any store. You want x to be as small as possible, i.e., you want to minimize the maximum number of products that are given to any store.

Return the minimum possible x. Example 1: Input: n = 6, quantities = [11,6] Output: 3 Explanation: One optimal way is: - The 11 products of type 0 are distributed to the first four stores in these amounts: 2, 3, 3, 3 - The 6 products of type 1 are distributed to the other two stores in these amounts: 3, 3 The maximum number of products given to any store is max(2, 3, 3, 3, 3, 3) = 3.

Example 2: Input: n = 7, quantities = [15,10,10] Output: 5 Explanation: One optimal way is: - The 15 products of type 0 are distributed to the first three stores in these amounts: 5, 5, 5 - The 10 products of type 1 are distributed to the next two stores in these amounts: 5, 5 - The 10 products of type 2 are distributed to the last two stores in these amounts: 5, 5 The maximum number of products given to any store is max(5, 5, 5, 5, 5, 5, 5) = 5.

Example 3: Input: n = 1, quantities = [100000] Output: 100000 Explanation: The only optimal way is: - The 100000 products of type 0 are distributed to the only store. The maximum number of products given to any store is max(100000) = 100000.

Constraints:

m == quantities.length 1 <= m <= n <= 105 1 <= quantities[i] <= 105

==

Problem Number: 1380 URL: https://leetcode.com/problems/walking-robot-simulation-ii Title: 2069. Walking Robot Simulation II Problem Description: A width x height grid is on an XY-plane with the bottom-left cell at (0, 0) and the top-right cell at (width - 1, height - 1). The grid is aligned with the four cardinal directions ("North", "East", "South", and "West"). A robot is initially at cell (0, 0) facing direction "East". The robot can be instructed to move for a specific number of steps. For each step, it does the following.

Attempts to move forward one cell in the direction it is facing. If the cell the robot is moving to is out of bounds, the robot instead turns 90 degrees counterclockwise and retries the step.

After the robot finishes moving the number of steps required, it stops and awaits the next instruction. Implement the Robot class:

Robot(int width, int height) Initializes the width x height grid with the robot at (0, 0) facing "East". void step(int num) Instructs the robot to move forward num steps. int[] getPos() Returns the current cell the robot is at, as an array of length 2, [x, y]. String getDir() Returns the current direction of the robot, "North", "East", "South", or "West".

Example 1:

Input ["Robot", "step", "step", "getPos", "getDir", "step", "step", "step", "getPos", "getDir"] [[6, 3], [2], [2], [], [], [2], [1], [4], [], []] Output [null, null, null, [4, 0], "East", null, null, null, [1, 2], "West"]

Explanation Robot robot = new Robot(6, 3); // Initialize the grid and the robot at (0, 0) facing East. robot.step(2); // It moves two steps East to (2, 0), and faces East. robot.step(2); // It moves two steps East to (4, 0), and faces East. robot.getPos(); // return [4, 0] robot.getDir(); // return "East" robot.step(2);

// It moves one step East to (5, 0), and faces East. // Moving the next step East would be out of bounds, so it turns and faces North. // Then, it moves one step North to (5, 1), and faces North. robot.step(1); // It moves one step North to (5, 2), and faces North (not West). robot.step(4); // Moving the next step North would be out of bounds, so it turns and faces West. // Then, it moves four steps West to (1, 2), and faces West. robot.getPos(); // return [1, 2] robot.getDir(); // return "West"

Constraints:

2 <= width, height <= 100 1 <= num <= 105 At most 104 calls in total will be made to step, getPos, and getDir.

==
Problem Number: 1381 URL: https://leetcode.com/problems/most-beautiful-item-for-each-query Title: 2070. Most Beautiful Item for Each Query Problem Description: You are given a 2D integer array items where items[i] = [pricei, beautyi] denotes the price and beauty of an item respectively. You are also given a 0-indexed integer array queries. For each queries[j], you want to determine the maximum beauty of an item whose price is less than or equal to queries[j]. If no such item exists, then the answer to this query is 0. Return an array answer of the same length as queries where answer[j] is the answer to the jth query. Example 1: Input: items = [[1,2],[3,2],[2,4],[5,6],[3,5]], queries = [1,2,3,4,5,6] Output: [2,4,5,5,6,6] Explanation: - For queries[0]=1, [1,2] is the only item which has price <= 1. Hence, the answer for this query is 2. - For queries[1]=2, the items which can be considered are [1,2] and [2,4]. The maximum beauty among them is 4. - For queries[2]=3 and queries[3]=4, the items which can be considered are [1,2], [3,2], [2,4], and [3,5]. The maximum beauty among them is 5. - For queries[4]=5 and queries[5]=6, all items can be considered. Hence, the answer for them is the maximum beauty of all items, i.e., 6.

Example 2: Input: items = [[1,2],[1,2],[1,3],[1,4]], queries = [1] Output: [4] Explanation: The price of every item is equal to 1, so we choose the item with the maximum beauty 4. Note that multiple items can have the same price and/or beauty.

Example 3: Input: items = [[10,1000]], queries = [5] Output: [0] Explanation: No item has a price less than or equal to 5, so no item can be chosen. Hence, the answer to the query is 0.

Constraints:

1 <= items.length, queries.length <= 105 items[i].length == 2 1 <= pricei, beautyi, queries[j] <= 109

==
Problem Number: 1382 URL: https://leetcode.com/problems/reverse-nodes-in-even-length-groups Title: 2074. Reverse Nodes in Even Length Groups Problem Description: You are given the head of a linked list. The nodes in the

linked list are sequentially assigned to non-empty groups whose lengths form the sequence of the natural numbers (1, 2, 3, 4, ...). The length of a group is the number of nodes assigned to it. In other words,

The 1st node is assigned to the first group. The 2nd and the 3rd nodes are assigned to the second group. The 4th, 5th, and 6th nodes are assigned to the third group, and so on.

Note that the length of the last group may be less than or equal to 1 + the length of the second to last group. Reverse the nodes in each group with an even length, and return the head of the modified linked list. Example 1:

Input: head = [5,2,6,3,9,1,7,3,8,4] Output: [5,6,2,3,9,1,4,8,3,7] Explanation: - The length of the first group is 1, which is odd, hence no reversal occurs. - The length of the second group is 2, which is even, hence the nodes are reversed. - The length of the third group is 3, which is odd, hence no reversal occurs. - The length of the last group is 4, which is even, hence the nodes are reversed.

Example 2:

Input: head = [1,1,0,6] Output: [1,0,1,6] Explanation: - The length of the first group is 1. No reversal occurs. - The length of the second group is 2. The nodes are reversed. - The length of the last group is 1. No reversal occurs.

Example 3:

Input: head = [1,1,0,6,5] Output: [1,0,1,5,6] Explanation: - The length of the first group is 1. No reversal occurs. - The length of the second group is 2. The nodes are reversed. - The length of the last group is 2. The nodes are reversed.

Constraints:

The number of nodes in the list is in the range [1, 105]. 0 <= Node.val <= 105

==

Problem Number: 1383 URL: https://leetcode.com/problems/decode-the-slanted-ciphertext Title: 2075. Decode the Slanted Ciphertext Problem Description: A string originalText is encoded using a slanted transposition cipher to a string encodedText with the help of a matrix having a fixed number of rows rows. originalText is placed first in a top-left to bottom-right manner.

The blue cells are filled first, followed by the red cells, then the yellow cells, and so on, until we reach the end of originalText. The arrow indicates the order in which the cells are filled. All empty cells are filled with ' '. The number of columns is chosen such that the rightmost column will not be empty after filling in originalText. encodedText is then formed by appending all characters of the matrix in a row-wise fashion.

The characters in the blue cells are appended first to encodedText, then the red cells, and so on, and finally the yellow cells. The arrow indicates the order in which the cells are accessed. For example, if originalText = "cipher" and rows = 3, then we encode it in the following manner:

The blue arrows depict how originalText is placed in the matrix, and the red arrows denote the order in which encodedText is formed. In the above example, encodedText = "ch ie pr". Given the encoded string encodedText and number of rows rows, return the original string originalText. Note: originalText does not have any trailing spaces ' '. The test cases are generated such that there is only one possible originalText. Example 1: Input: encodedText = "ch ie pr", rows = 3 Output: "cipher" Explanation: This is the same example described in the problem description.

Example 2:

Input: encodedText = "iveo eed l te olc", rows = 4 Output: "i love leetcode" Explanation: The figure above denotes the matrix that was used to encode originalText. The blue arrows show how we can find originalText from encodedText.

Example 3:

Input: encodedText = "coding", rows = 1 Output: "coding" Explanation: Since there is only 1 row, both originalText and encodedText are the same.

Constraints:

0 <= encodedText.length <= 106 encodedText consists of lowercase English letters and ' ' only. encodedText is a valid encoding of some originalText that does not have trailing spaces. 1 <= rows <= 1000 The testcases are generated such that there is only one possible originalText.

==
Problem Number: 1384 URL: https://leetcode.com/problems/watering-plants Title: 2079. Watering Plants Problem Description: You want to water n plants in your garden with a watering can. The plants are arranged in a row and are labeled from 0 to n - 1 from left to right where the ith plant is located at x = i. There is a river at x = -1 that you can refill your watering can at. Each plant needs a specific amount of water. You will water the plants in the following way:

Water the plants in order from left to right. After watering the current plant, if you do not have enough water to completely water the next plant, return to the river to fully refill the watering can. You cannot refill the watering can early.

You are initially at the river (i.e., x = -1). It takes one step to move one unit on the x-axis. Given a 0-indexed integer array plants of n integers, where plants[i] is the amount of water the ith plant needs, and an integer capacity representing the watering can capacity, return the number of steps needed to water all the plants. Example 1: Input: plants = [2,2,3,3], capacity = 5 Output: 14 Explanation: Start at the river with a full watering can: - Walk to plant 0 (1 step) and water it. Watering can has 3 units of water. - Walk to plant 1 (1 step) and water it. Watering can has 1 unit of water. - Since you cannot completely water plant 2, walk back to the river to refill (2 steps). - Walk to plant 2 (3 steps) and water it. Watering can has 2 units of water. - Since you cannot completely water plant 3, walk back to the river to refill (3 steps). - Walk to plant 3 (4 steps) and water it. Steps needed = 1 + 1 + 2 + 3 + 3 + 4 = 14.

Example 2: Input: plants = [1,1,1,4,2,3], capacity = 4 Output: 30 Explanation: Start at the river with a full watering can: - Water plants 0, 1, and 2 (3 steps). Return to river (3 steps). - Water plant 3 (4 steps). Return to river (4 steps). - Water plant 4 (5 steps). Return to river (5 steps). - Water plant 5 (6 steps). Steps needed = 3 + 3 + 4 + 4 + 5 + 5 + 6 = 30.

Example 3: Input: plants = [7,7,7,7,7,7,7], capacity = 8 Output: 49 Explanation: You have to refill before watering each plant. Steps needed = 1 + 1 + 2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 + 7 = 49.

Constraints:

n == plants.length 1 <= n <= 1000 1 <= plants[i] <= 106 max(plants[i]) <= capacity <= 109

==

Problem Number: 1385 URL: https://leetcode.com/problems/range-frequency-queries Title: 2080. Range Frequency Queries Problem Description: Design a data structure to find the frequency of a given value in a given subarray. The frequency of a value in a subarray is the number of occurrences of that value in the subarray. Implement the RangeFreqQuery class:

RangeFreqQuery(int[] arr) Constructs an instance of the class with the given 0-indexed integer array arr. int query(int left, int right, int value) Returns the frequency of value in the subarray arr[left...right].

A subarray is a contiguous sequence of elements within an array. arr[left...right] denotes the subarray that contains the elements of nums between indices left and right (inclusive). Example 1: Input ["RangeFreqQuery", "query", "query"] [[[12, 33, 4, 56, 22, 2, 34, 33, 22, 12, 34, 56]], [1, 2, 4], [0, 11, 33]] Output [null, 1, 2]

Explanation RangeFreqQuery rangeFreqQuery = new RangeFreqQuery([12, 33, 4, 56, 22, 2, 34, 33, 22, 12, 34, 56]); rangeFreqQuery.query(1, 2, 4); // return 1. The value 4 occurs 1 time in the subarray [33, 4] rangeFreqQuery.query(0, 11, 33); // return 2. The value 33 occurs 2 times in the whole array.

Constraints:

1 <= arr.length <= 105 1 <= arr[i], value <= 104 0 <= left <= right < arr.length At most 105 calls will be made to query

==

Problem Number: 1386 URL: https://leetcode.com/problems/minimum-number-of-food-buckets-to-feed-the-hamsters Title: 2086. Minimum Number of Food Buckets to Feed the Hamsters Problem Description: You are given a 0-indexed string hamsters where hamsters[i] is either:

'H' indicating that there is a hamster at index i, or '.' indicating that index i is empty.

You will add some number of food buckets at the empty indices in order to feed the hamsters. A hamster can be fed if there is at least one food bucket to its left or to its right. More formally, a hamster at index i can be fed if you place a food bucket at index i - 1 and/or at index i + 1. Return the minimum number of food buckets you should place at empty indices to feed all the hamsters or -1 if it is impossible to feed all of them. Example 1:

Input: hamsters = "H..H" Output: 2 Explanation: We place two food buckets at indices 1 and 2. It can be shown that if we place only one food bucket, one of the hamsters will not be fed.

Example 2:

Input: hamsters = ".H.H." Output: 1 Explanation: We place one food bucket at index 2.

Example 3:

Input: hamsters = ".HHH." Output: -1 Explanation: If we place a food bucket at every empty index as shown, the hamster at index 2 will not be able to eat.

Constraints:

1 <= hamsters.length <= 105 hamsters[i] is either'H' or '.'.

==

Problem Number: 1387 URL: https://leetcode.com/problems/minimum-cost-homecoming-of-a-robot-in-a-grid Title: 2087. Minimum Cost Homecoming of a Robot in a Grid Problem Description: There is an m x n grid, where (0, 0) is the top-left cell and (m - 1, n - 1) is the bottom-right cell. You are given an integer array startPos where startPos = [startrow, startcol] indicates that initially, a robot is at the cell (startrow, startcol). You are also given an integer array homePos where homePos = [homerow, homecol] indicates that its home is at the cell (homerow, homecol). The robot needs to go to its home. It can move one cell in four directions: left, right, up, or down, and it can not move outside the boundary. Every move incurs some cost. You are further given two 0-indexed integer arrays: rowCosts of length m and colCosts of length n.

If the robot moves up or down into a cell whose row is r, then this move costs rowCosts[r]. If the robot moves left or right into a cell whose column is c, then this move costs colCosts[c].

Return the minimum total cost for this robot to return home. Example 1:

Input: startPos = [1, 0], homePos = [2, 3], rowCosts = [5, 4, 3], colCosts = [8, 2, 6, 7] Output: 18 Explanation: One optimal path is that: Starting from (1, 0) -> It goes down to (2, 0). This move costs rowCosts[2] = 3. -> It goes right to (2, 1). This move costs colCosts[1] = 2. -> It goes right to (2, 2). This move costs colCosts[2] = 6. -> It goes right to (2, 3). This move costs colCosts[3] =

7. The total cost is 3 + 2 + 6 + 7 = 18 Example 2: Input: startPos = [0, 0], homePos = [0, 0], rowCosts = [5], colCosts = [26] Output: 0 Explanation: The robot is already at its home. Since no moves occur, the total cost is 0.

Constraints:

m == rowCosts.length n == colCosts.length 1 <= m, n <= 105 0 <= rowCosts[r], colCosts[c] <= 104 startPos.length == 2 homePos.length == 2 0 <= startrow, homerow < m 0 <= startcol, homecol < n

==

Problem Number: 1388 URL: https://leetcode.com/problems/k-radius-subarray-averages Title: 2090. K Radius Subarray Averages Problem Description: You are given a 0-indexed array nums of n integers, and an integer k. The k-radius average for a subarray of nums centered at some index i with the radius k is the average of all elements in nums between the indices i - k and i + k (inclusive). If there are less than k elements before or after the index i, then the k-radius average is -1. Build and return an array avgs of length n where avgs[i] is the k-radius average for the subarray centered at index i. The average of x elements is the sum of the x elements divided by x, using integer division. The integer division truncates toward zero, which means losing its fractional part.

For example, the average of four elements 2, 3, 1, and 5 is (2 + 3 + 1 + 5) / 4 = 11 / 4 = 2.75, which truncates to 2.

Example 1:

Input: nums = [7,4,3,9,1,8,5,2,6], k = 3 Output: [-1,-1,-1,5,4,4,-1,-1,-1] Explanation: - avg[0], avg[1], and avg[2] are -1 because there are less than k elements before each index. - The sum of the subarray centered at index 3 with radius 3 is: 7 + 4 + 3 + 9 + 1 + 8 + 5 = 37. Using integer division, avg[3] = 37 / 7 = 5. - For the subarray centered at index 4, avg[4] = (4 + 3 + 9 + 1 + 8 + 5 + 2) / 7 = 4. - For the subarray centered at index 5, avg[5] = (3 + 9 + 1 + 8 + 5 + 2 + 6) / 7 = 4. - avg[6], avg[7], and avg[8] are -1 because there are less than k elements after each index.

Example 2: Input: nums = [100000], k = 0 Output: [100000] Explanation: - The sum of the subarray centered at index 0 with radius 0 is: 100000. avg[0] = 100000 / 1 = 100000.

Example 3: Input: nums = [8], k = 100000 Output: [-1] Explanation: - avg[0] is -1 because there are less than k elements before and after index 0.

Constraints:

n == nums.length 1 <= n <= 105 0 <= nums[i], k <= 105

==

Problem Number: 1389 URL: https://leetcode.com/problems/removing-minimum-and-maximum-from-array Title: 2091. Removing Minimum and

Maximum From Array Problem Description: You are given a 0-indexed array of distinct integers nums. There is an element in nums that has the lowest value and an element that has the highest value. We call them the minimum and maximum respectively. Your goal is to remove both these elements from the array. A deletion is defined as either removing an element from the front of the array or removing an element from the back of the array. Return the minimum number of deletions it would take to remove both the minimum and maximum element from the array.   Example 1: Input: nums = [2,10,7,5,4,1,8,6] Output: 5 Explanation: The minimum element in the array is nums[5], which is 1. The maximum element in the array is nums[1], which is 10. We can remove both the minimum and maximum by removing 2 elements from the front and 3 elements from the back. This results in 2 + 3 = 5 deletions, which is the minimum number possible.

Example 2: Input: nums = [0,-4,19,1,8,-2,-3,5] Output: 3 Explanation: The minimum element in the array is nums[1], which is -4. The maximum element in the array is nums[2], which is 19. We can remove both the minimum and maximum by removing 3 elements from the front. This results in only 3 deletions, which is the minimum number possible.

Example 3: Input: nums = [101] Output: 1 Explanation: There is only one element in the array, which makes it both the minimum and maximum element. We can remove it with 1 deletion.

Constraints:

1 <= nums.length <= 105 -105 <= nums[i] <= 105 The integers in nums are distinct.

==

Problem Number: 1390 URL: https://leetcode.com/problems/delete-the-middle-node-of-a-linked-list Title: 2095. Delete the Middle Node of a Linked List Problem Description: You are given the head of a linked list. Delete the middle node, and return the head of the modified linked list. The middle node of a linked list of size n is the n / 2 th node from the start using 0-based indexing, where x denotes the largest integer less than or equal to x.

For n = 1, 2, 3, 4, and 5, the middle nodes are 0, 1, 1, 2, and 2, respectively.

Example 1:

Input: head = [1,3,4,7,1,2,6] Output: [1,3,4,1,2,6] Explanation: The above figure represents the given linked list. The indices of the nodes are written below. Since n = 7, node 3 with value 7 is the middle node, which is marked in red. We return the new list after removing this node.

Example 2:

Input: head = [1,2,3,4] Output: [1,2,4] Explanation: The above figure represents the given linked list. For n = 4, node 2 with value 3 is the middle node, which is marked in red.

Example 3:

Input: head = [2,1] Output: [2] Explanation: The above figure represents the given linked list. For n = 2, node 1 with value 1 is the middle node, which is marked in red. Node 0 with value 2 is the only node remaining after removing node 1. Constraints:

The number of nodes in the list is in the range [1, 105]. 1 <= Node.val <= 105

==
Problem Number: 1391 URL: https://leetcode.com/problems/step-by-step-directions-from-a-binary-tree-node-to-another Title: 2096. Step-By-Step Directions From a Binary Tree Node to Another Problem Description: You are given the root of a binary tree with n nodes. Each node is uniquely assigned a value from 1 to n. You are also given an integer startValue representing the value of the start node s, and a different integer destValue representing the value of the destination node t. Find the shortest path starting from node s and ending at node t. Generate step-by-step directions of such path as a string consisting of only the uppercase letters 'L', 'R', and 'U'. Each letter indicates a specific direction:

'L' means to go from a node to its left child node. 'R' means to go from a node to its right child node. 'U' means to go from a node to its parent node.

Return the step-by-step directions of the shortest path from node s to node t. Example 1:

Input: root = [5,1,2,3,null,6,4], startValue = 3, destValue = 6 Output: "UURL" Explanation: The shortest path is: 3 → 1 → 5 → 2 → 6.

Example 2:

Input: root = [2,1], startValue = 2, destValue = 1 Output: "L" Explanation: The shortest path is: 2 → 1.

Constraints:

The number of nodes in the tree is n. 2 <= n <= 105 1 <= Node.val <= n All the values in the tree are unique. 1 <= startValue, destValue <= n startValue != destValue

==
Problem Number: 1392 URL: https://leetcode.com/problems/find-good-days-to-rob-the-bank Title: 2100. Find Good Days to Rob the Bank Problem Description: You and a gang of thieves are planning on robbing a bank. You are given a 0-indexed integer array security, where security[i] is the number of guards on duty on the ith day. The days are numbered starting from 0. You are also given an integer time. The ith day is a good day to rob the bank if:

There are at least time days before and after the ith day, The number of guards at the bank for the time days before i are non-increasing, and The number of guards at the bank for the time days after i are non-decreasing.

More formally, this means day i is a good day to rob the bank if and only if security[i - time] >= security[i - time + 1] >= ... >= security[i] <= ... <= security[i + time - 1] <= security[i + time]. Return a list of all days (0-indexed) that are good days to rob the bank. The order that the days are returned in does not matter.   Example 1: Input: security = [5,3,3,3,5,6,2], time = 2 Output: [2,3] Explanation: On day 2, we have security[0] >= security[1] >= security[2] <= security[3] <= security[4]. On day 3, we have security[1] >= security[2] >= security[3] <= security[4] <= security[5]. No other days satisfy this condition, so days 2 and 3 are the only good days to rob the bank.

Example 2: Input: security = [1,1,1,1,1], time = 0 Output: [0,1,2,3,4] Explanation: Since time equals 0, every day is a good day to rob the bank, so return every day.

Example 3: Input: security = [1,2,3,4,5,6], time = 2 Output: [] Explanation: No day has 2 days before it that have a non-increasing number of guards. Thus, no day is a good day to rob the bank, so return an empty list.

Constraints:

1 <= security.length <= 105 0 <= security[i], time <= 105

==
Problem Number: 1393 URL: https://leetcode.com/problems/detonate-the-maximum-bombs Title: 2101. Detonate the Maximum Bombs Problem Description: You are given a list of bombs. The range of a bomb is defined as the area where its effect can be felt. This area is in the shape of a circle with the center as the location of the bomb. The bombs are represented by a 0-indexed 2D integer array bombs where bombs[i] = [xi, yi, ri]. xi and yi denote the X-coordinate and Y-coordinate of the location of the ith bomb, whereas ri denotes the radius of its range. You may choose to detonate a single bomb. When a bomb is detonated, it will detonate all bombs that lie in its range. These bombs will further detonate the bombs that lie in their ranges. Given the list of bombs, return the maximum number of bombs that can be detonated if you are allowed to detonate only one bomb.   Example 1:

Input: bombs = [[2,1,3],[6,1,4]] Output: 2 Explanation: The above figure shows the positions and ranges of the 2 bombs. If we detonate the left bomb, the right bomb will not be affected. But if we detonate the right bomb, both bombs will be detonated. So the maximum bombs that can be detonated is max(1, 2) = 2.

Example 2:

Input: bombs = [[1,1,5],[10,10,5]] Output: 1 Explanation: Detonating either bomb will not detonate the other bomb, so the maximum number of bombs that can be detonated is 1.

Example 3:

Input: bombs = [[1,2,3],[2,3,1],[3,4,2],[4,5,3],[5,6,4]] Output: 5 Explanation: The best bomb to detonate is bomb 0 because: - Bomb 0 detonates bombs 1 and 2.

The red circle denotes the range of bomb 0. - Bomb 2 detonates bomb 3. The blue circle denotes the range of bomb 2. - Bomb 3 detonates bomb 4. The green circle denotes the range of bomb 3. Thus all 5 bombs are detonated.

Constraints:

1 <= bombs.length <= 100 bombs[i].length == 3 1 <= xi, yi, ri <= 105

==

Problem Number: 1394 URL: https://leetcode.com/problems/sum-of-subarray-ranges Title: 2104. Sum of Subarray Ranges Problem Description: You are given an integer array nums. The range of a subarray of nums is the difference between the largest and smallest element in the subarray. Return the sum of all subarray ranges of nums. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,2,3] Output: 4 Explanation: The 6 subarrays of nums are the following: [1], range = largest - smallest = 1 - 1 = 0 [2], range = 2 - 2 = 0 [3], range = 3 - 3 = 0 [1,2], range = 2 - 1 = 1 [2,3], range = 3 - 2 = 1 [1,2,3], range = 3 - 1 = 2 So the sum of all ranges is 0 + 0 + 0 + 1 + 1 + 2 = 4. Example 2: Input: nums = [1,3,3] Output: 4 Explanation: The 6 subarrays of nums are the following: [1], range = largest - smallest = 1 - 1 = 0 [3], range = 3 - 3 = 0 [3], range = 3 - 3 = 0 [1,3], range = 3 - 1 = 2 [3,3], range = 3 - 3 = 0 [1,3,3], range = 3 - 1 = 2 So the sum of all ranges is 0 + 0 + 0 + 2 + 0 + 2 = 4.

Example 3: Input: nums = [4,-2,-3,4,1] Output: 59 Explanation: The sum of all subarray ranges of nums is 59.

Constraints:

1 <= nums.length <= 1000 -109 <= nums[i] <= 109

Follow-up: Could you find a solution with O(n) time complexity?

==

Problem Number: 1395 URL: https://leetcode.com/problems/watering-plants-ii Title: 2105. Watering Plants II Problem Description: Alice and Bob want to water n plants in their garden. The plants are arranged in a row and are labeled from 0 to n - 1 from left to right where the ith plant is located at x = i. Each plant needs a specific amount of water. Alice and Bob have a watering can each, initially full. They water the plants in the following way:

Alice waters the plants in order from left to right, starting from the 0th plant. Bob waters the plants in order from right to left, starting from the (n - 1)th plant. They begin watering the plants simultaneously. It takes the same amount of time to water each plant regardless of how much water it needs. Alice/Bob must water the plant if they have enough in their can to fully water it. Otherwise, they first refill their can (instantaneously) then water the plant. In case both Alice and Bob reach the same plant, the one with more water currently in his/her watering can should water this plant. If they have the same amount of water, then Alice should water this plant.

Given a 0-indexed integer array plants of n integers, where plants[i] is the amount of water the ith plant needs, and two integers capacityA and capacityB representing the capacities of Alice's and Bob's watering cans respectively, return the number of times they have to refill to water all the plants. Example 1: Input: plants = [2,2,3,3], capacityA = 5, capacityB = 5 Output: 1 Explanation: - Initially, Alice and Bob have 5 units of water each in their watering cans. - Alice waters plant 0, Bob waters plant 3. - Alice and Bob now have 3 units and 2 units of water respectively. - Alice has enough water for plant 1, so she waters it. Bob does not have enough water for plant 2, so he refills his can then waters it. So, the total number of times they have to refill to water all the plants is 0 + 0 + 1 + 0 = 1.

Example 2: Input: plants = [2,2,3,3], capacityA = 3, capacityB = 4 Output: 2 Explanation: - Initially, Alice and Bob have 3 units and 4 units of water in their watering cans respectively. - Alice waters plant 0, Bob waters plant 3. - Alice and Bob now have 1 unit of water each, and need to water plants 1 and 2 respectively. - Since neither of them have enough water for their current plants, they refill their cans and then water the plants. So, the total number of times they have to refill to water all the plants is 0 + 1 + 1 + 0 = 2.

Example 3: Input: plants = [5], capacityA = 10, capacityB = 8 Output: 0 Explanation: - There is only one plant. - Alice's watering can has 10 units of water, whereas Bob's can has 8 units. Since Alice has more water in her can, she waters this plant. So, the total number of times they have to refill is 0.

Constraints:

n == plants.length 1 <= n <= 105 1 <= plants[i] <= 106 max(plants[i]) <= capacityA, capacityB <= 109

==

Problem Number: 1396 URL: https://leetcode.com/problems/adding-spaces-to-a-string Title: 2109. Adding Spaces to a String Problem Description: You are given a 0-indexed string s and a 0-indexed integer array spaces that describes the indices in the original string where spaces will be added. Each space should be inserted before the character at the given index.

For example, given s = "EnjoyYourCoffee" and spaces = [5, 9], we place spaces before 'Y' and 'C', which are at indices 5 and 9 respectively. Thus, we obtain "Enjoy Your Coffee".

Return the modified string after the spaces have been added. Example 1: Input: s = "LeetcodeHelpsMeLearn", spaces = [8,13,15] Output: "Leetcode Helps Me Learn" Explanation: The indices 8, 13, and 15 correspond to the underlined characters in "LeetcodeHelpsMeLearn". We then place spaces before those characters.

Example 2: Input: s = "icodeinpython", spaces = [1,5,7,9] Output: "i code in py thon" Explanation: The indices 1, 5, 7, and 9 correspond to the underlined characters in "icodeinpython". We then place spaces before those characters.

Example 3: Input: s = ”spacing”, spaces = [0,1,2,3,4,5,6] Output: ” s p a c i n g” Explanation: We are also able to place spaces before the first character of the string.

Constraints:

1 <= s.length <= 3 * 105 s consists only of lowercase and uppercase English letters. 1 <= spaces.length <= 3 * 105 0 <= spaces[i] <= s.length - 1 All the values of spaces are strictly increasing.

==
Problem Number: 1397 URL: https://leetcode.com/problems/number-of-smooth-descent-periods-of-a-stock Title: 2110. Number of Smooth Descent Periods of a Stock Problem Description: You are given an integer array prices representing the daily price history of a stock, where prices[i] is the stock price on the ith day. A smooth descent period of a stock consists of one or more contiguous days such that the price on each day is lower than the price on the preceding day by exactly 1. The first day of the period is exempted from this rule. Return the number of smooth descent periods. Example 1: Input: prices = [3,2,1,4] Output: 7 Explanation: There are 7 smooth descent periods: [3], [2], [1], [4], [3,2], [2,1], and [3,2,1] Note that a period with one day is a smooth descent period by the definition.

Example 2: Input: prices = [8,6,7,7] Output: 4 Explanation: There are 4 smooth descent periods: [8], [6], [7], and [7] Note that [8,6] is not a smooth descent period as 8 - 6 1.

Example 3: Input: prices = [1] Output: 1 Explanation: There is 1 smooth descent period: [1]

Constraints:

1 <= prices.length <= 105 1 <= prices[i] <= 105

==
Problem Number: 1398 URL: https://leetcode.com/problems/find-all-possible-recipes-from-given-supplies Title: 2115. Find All Possible Recipes from Given Supplies Problem Description: You have information about n different recipes. You are given a string array recipes and a 2D string array ingredients. The ith recipe has the name recipes[i], and you can create it if you have all the needed ingredients from ingredients[i]. Ingredients to a recipe may need to be created from other recipes, i.e., ingredients[i] may contain a string that is in recipes. You are also given a string array supplies containing all the ingredients that you initially have, and you have an infinite supply of all of them. Return a list of all the recipes that you can create. You may return the answer in any order. Note that two recipes may contain each other in their ingredients. Example 1: Input: recipes = [”bread”], ingredients = [[”yeast”,”flour”]], supplies = [”yeast”,”flour”,”corn”] Output: [”bread”] Explanation: We can create ”bread” since we have the ingredients ”yeast” and ”flour”.

Example 2: Input: recipes = ["bread","sandwich"], ingredients = [["yeast","flour"],["bread","meat"]], supplies = ["yeast","flour","meat"] Output: ["bread","sandwich"] Explanation: We can create "bread" since we have the ingredients "yeast" and "flour". We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread".

Example 3: Input: recipes = ["bread","sandwich","burger"], ingredients = [["yeast","flour"],["bread","meat"],["sandwich","meat","bread"]], supplies = ["yeast","flour","meat"] Output: ["bread","sandwich","burger"] Explanation: We can create "bread" since we have the ingredients "yeast" and "flour". We can create "sandwich" since we have the ingredient "meat" and can create the ingredient "bread". We can create "burger" since we have the ingredient "meat" and can create the ingredients "bread" and "sandwich".

Constraints:

n == recipes.length == ingredients.length 1 <= n <= 100 1 <= ingredients[i].length, supplies.length <= 100 1 <= recipes[i].length, ingredients[i][j].length, supplies[k].length <= 10 recipes[i], ingredients[i][j], and supplies[k] consist only of lowercase English letters. All the values of recipes and supplies combined are unique. Each ingredients[i] does not contain any duplicate values.

==

Problem Number: 1399 URL: https://leetcode.com/problems/check-if-a-parentheses-string-can-be-valid Title: 2116. Check if a Parentheses String Can Be Valid Problem Description: A parentheses string is a non-empty string consisting only of '(' and ')'. It is valid if any of the following conditions is true:

It is (). It can be written as AB (A concatenated with B), where A and B are valid parentheses strings. It can be written as (A), where A is a valid parentheses string.

You are given a parentheses string s and a string locked, both of length n. locked is a binary string consisting only of '0's and '1's. For each index i of locked,

If locked[i] is '1', you cannot change s[i]. But if locked[i] is '0', you can change s[i] to either '(' or ')'.

Return true if you can make s a valid parentheses string. Otherwise, return false. Example 1:

Input: s = "))()))", locked = "010100" Output: true Explanation: locked[1] == '1' and locked[3] == '1', so we cannot change s[1] or s[3]. We change s[0] and s[4] to '(' while leaving s[2] and s[5] unchanged to make s valid. Example 2: Input: s = "()()", locked = "0000" Output: true Explanation: We do not need to make any changes because s is already valid.

Example 3: Input: s = ")", locked = "0" Output: false Explanation: locked permits us to change s[0]. Changing s[0] to either '(' or ')' will not make s valid.

Constraints:

n == s.length == locked.length 1 <= n <= 105 s[i] is either '(' or ')'. locked[i] is either '0' or '1'.

==
Problem Number: 1400 URL: https://leetcode.com/problems/execution-of-all-suffix-instructions-staying-in-a-grid Title: 2120. Execution of All Suffix Instructions Staying in a Grid Problem Description: There is an n x n grid, with the top-left cell at (0, 0) and the bottom-right cell at (n - 1, n - 1). You are given the integer n and an integer array startPos where startPos = [startrow, startcol] indicates that a robot is initially at cell (startrow, startcol). You are also given a 0-indexed string s of length m where s[i] is the ith instruction for the robot: 'L' (move left), 'R' (move right), 'U' (move up), and 'D' (move down). The robot can begin executing from any ith instruction in s. It executes the instructions one by one towards the end of s but it stops if either of these conditions is met:

The next instruction will move the robot off the grid. There are no more instructions left to execute.

Return an array answer of length m where answer[i] is the number of instructions the robot can execute if the robot begins executing from the ith instruction in s. Example 1:

Input: n = 3, startPos = [0,1], s = "RRDDLU" Output: [1,5,4,3,1,0] Explanation: Starting from startPos and beginning execution from the ith instruction: - 0th: "RRDDLU". Only one instruction "R" can be executed before it moves off the grid. - 1st: "RDDLU". All five instructions can be executed while it stays in the grid and ends at (1, 1). - 2nd: "DDLU". All four instructions can be executed while it stays in the grid and ends at (1, 0). - 3rd: "DLU". All three instructions can be executed while it stays in the grid and ends at (0, 0). - 4th: "LU". Only one instruction "L" can be executed before it moves off the grid. - 5th: "U". If moving up, it would move off the grid.

Example 2:

Input: n = 2, startPos = [1,1], s = "LURD" Output: [4,1,0,0] Explanation: - 0th: "LURD". - 1st: "URD". - 2nd: "RD". - 3rd: "D".

Example 3:

Input: n = 1, startPos = [0,0], s = "LRUD" Output: [0,0,0,0] Explanation: No matter which instruction the robot begins execution from, it would move off the grid.

Constraints:

m == s.length 1 <= n, m <= 500 startPos.length == 2 0 <= startrow, startcol < n s consists of 'L', 'R', 'U', and 'D'.

==
Problem Number: 1401 URL: https://leetcode.com/problems/intervals-between-identical-elements Title: 2121. Intervals Between Identical Elements Problem Description: You are given a 0-indexed array of n integers arr. The interval between two elements in arr is defined as the absolute difference between their indices. More formally, the interval between arr[i] and arr[j] is |i - j|. Return an array intervals of length n where intervals[i] is the sum of intervals between arr[i] and each element in arr with the same value as arr[i]. Note: |x| is the absolute value of x. Example 1: Input: arr = [2,1,3,1,2,3,3] Output: [4,2,7,2,4,4,5] Explanation: - Index 0: Another 2 is found at index 4. |0 - 4| = 4 - Index 1: Another 1 is found at index 3. |1 - 3| = 2 - Index 2: Two more 3s are found at indices 5 and 6. |2 - 5| + |2 - 6| = 7 - Index 3: Another 1 is found at index 1. |3 - 1| = 2 - Index 4: Another 2 is found at index 0. |4 - 0| = 4 - Index 5: Two more 3s are found at indices 2 and 6. |5 - 2| + |5 - 6| = 4 - Index 6: Two more 3s are found at indices 2 and 5. |6 - 2| + |6 - 5| = 5

Example 2: Input: arr = [10,5,10,10] Output: [5,0,3,4] Explanation: - Index 0: Two more 10s are found at indices 2 and 3. |0 - 2| + |0 - 3| = 5 - Index 1: There is only one 5 in the array, so its sum of intervals to identical elements is 0. - Index 2: Two more 10s are found at indices 0 and 3. |2 - 0| + |2 - 3| = 3 - Index 3: Two more 10s are found at indices 0 and 2. |3 - 0| + |3 - 2| = 4

Constraints:

n == arr.length 1 <= n <= 105 1 <= arr[i] <= 105
==
Problem Number: 1402 URL: https://leetcode.com/problems/number-of-laser-beams-in-a-bank Title: 2125. Number of Laser Beams in a Bank Problem Description: Anti-theft security devices are activated inside a bank. You are given a 0-indexed binary string array bank representing the floor plan of the bank, which is an m x n 2D matrix. bank[i] represents the ith row, consisting of '0's and '1's. '0' means the cell is empty, while'1' means the cell has a security device. There is one laser beam between any two security devices if both conditions are met:

The two devices are located on two different rows: r1 and r2, where r1 < r2. For each row i where r1 < i < r2, there are no security devices in the ith row.

Laser beams are independent, i.e., one beam does not interfere nor join with another. Return the total number of laser beams in the bank. Example 1:

Input: bank = ["011001","000000","010100","001000"] Output: 8 Explanation: Between each of the following device pairs, there is one beam. In total, there are 8 beams: * bank[0][1] -- bank[2][1] * bank[0][1] -- bank[2][3] * bank[0][2] -- bank[2][1] * bank[0][2] -- bank[2][3] * bank[0][5] -- bank[2][1] * bank[0][5] -- bank[2][3] * bank[2][1] -- bank[3][2] * bank[2][3] -- bank[3][2] Note that there is no beam between any device on the 0th row with any on the 3rd row. This is because the 2nd row contains security devices, which breaks the second condition.

Example 2:

Input: bank = ["000","111","000"] Output: 0 Explanation: There does not exist two devices located on two different rows.

Constraints:

m == bank.length n == bank[i].length 1 <= m, n <= 500 bank[i][j] is either '0' or '1'.

==

Problem Number: 1403 URL: https://leetcode.com/problems/destroying-asteroids Title: 2126. Destroying Asteroids Problem Description: You are given an integer mass, which represents the original mass of a planet. You are further given an integer array asteroids, where asteroids[i] is the mass of the ith asteroid. You can arrange for the planet to collide with the asteroids in any arbitrary order. If the mass of the planet is greater than or equal to the mass of the asteroid, the asteroid is destroyed and the planet gains the mass of the asteroid. Otherwise, the planet is destroyed. Return true if all asteroids can be destroyed. Otherwise, return false. Example 1: Input: mass = 10, asteroids = [3,9,19,5,21] Output: true Explanation: One way to order the asteroids is [9,19,5,3,21]: - The planet collides with the asteroid with a mass of 9. New planet mass: 10 + 9 = 19 - The planet collides with the asteroid with a mass of 19. New planet mass: 19 + 19 = 38 - The planet collides with the asteroid with a mass of 5. New planet mass: 38 + 5 = 43 - The planet collides with the asteroid with a mass of 3. New planet mass: 43 + 3 = 46 - The planet collides with the asteroid with a mass of 21. New planet mass: 46 + 21 = 67 All asteroids are destroyed.

Example 2: Input: mass = 5, asteroids = [4,9,23,4] Output: false Explanation: The planet cannot ever gain enough mass to destroy the asteroid with a mass of 23. After the planet destroys the other asteroids, it will have a mass of 5 + 4 + 9 + 4 = 22. This is less than 23, so a collision would not destroy the last asteroid. Constraints:

1 <= mass <= 105 1 <= asteroids.length <= 105 1 <= asteroids[i] <= 105

==

Problem Number: 1404 URL: https://leetcode.com/problems/maximum-twin-sum-of-a-linked-list Title: 2130. Maximum Twin Sum of a Linked List Problem Description: In a linked list of size n, where n is even, the ith node (0-indexed) of the linked list is known as the twin of the (n-1-i)th node, if 0 <= i <= (n / 2) - 1.

For example, if n = 4, then node 0 is the twin of node 3, and node 1 is the twin of node 2. These are the only nodes with twins for n = 4.

The twin sum is defined as the sum of a node and its twin. Given the head of a linked list with even length, return the maximum twin sum of the linked list. Example 1:

Input: head = [5,4,2,1] Output: 6 Explanation: Nodes 0 and 1 are the twins of nodes 3 and 2, respectively. All have twin sum = 6. There are no other nodes with twins in the linked list. Thus, the maximum twin sum of the linked list is 6.

Example 2:

Input: head = [4,2,2,3] Output: 7 Explanation: The nodes with twins present in this linked list are: - Node 0 is the twin of node 3 having a twin sum of 4 + 3 = 7. - Node 1 is the twin of node 2 having a twin sum of 2 + 2 = 4. Thus, the maximum twin sum of the linked list is max(7, 4) = 7.

Example 3:

Input: head = [1,100000] Output: 100001 Explanation: There is only one node with a twin in the linked list having twin sum of 1 + 100000 = 100001.

Constraints:

The number of nodes in the list is an even integer in the range [2, 105]. 1 <= Node.val <= 105

==
Problem Number: 1405 URL: https://leetcode.com/problems/longest-palindrome-by-concatenating-two-letter-words Title: 2131. Longest Palindrome by Concatenating Two Letter Words Problem Description: You are given an array of strings words. Each element of words consists of two lowercase English letters. Create the longest possible palindrome by selecting some elements from words and concatenating them in any order. Each element can be selected at most once. Return the length of the longest palindrome that you can create. If it is impossible to create any palindrome, return 0. A palindrome is a string that reads the same forward and backward. Example 1: Input: words = ["lc","cl","gg"] Output: 6 Explanation: One longest palindrome is "lc" + "gg" + "cl" = "lcggcl", of length 6. Note that "clgglc" is another longest palindrome that can be created.

Example 2: Input: words = ["ab","ty","yt","lc","cl","ab"] Output: 8 Explanation: One longest palindrome is "ty" + "lc" + "cl" + "yt" = "tylcclyt", of length 8. Note that "lcyttycl" is another longest palindrome that can be created.

Example 3: Input: words = ["cc","ll","xx"] Output: 2 Explanation: One longest palindrome is "cc", of length 2. Note that "ll" is another longest palindrome that can be created, and so is "xx".

Constraints:

1 <= words.length <= 105 words[i].length == 2 words[i] consists of lowercase English letters.

==
Problem Number: 1406 URL: https://leetcode.com/problems/minimum-swaps-to-group-all-1s-together-ii Title: 2134. Minimum Swaps to Group All 1's Together II Problem Description: A swap is defined as taking two distinct positions in an array and swapping the values in them. A circular array is defined as an array where we consider the first element and the last element to be adjacent. Given a binary circular array nums, return the minimum number of swaps required to group all 1's present in the array together at any location. Example 1: Input: nums = [0,1,0,1,1,0,0] Output: 1 Explanation: Here are a few of the ways to group all the 1's together: [0,0,1,1,1,0,0] using 1 swap. [0,1,1,1,0,0,0] using 1 swap. [1,1,0,0,0,0,1] using 2 swaps (using the circular property of the array). There is no way to group all 1's together with 0 swaps. Thus, the minimum number of swaps required is 1.

Example 2: Input: nums = [0,1,1,1,0,0,1,1,0] Output: 2 Explanation: Here are a few of the ways to group all the 1's together: [1,1,1,0,0,0,0,1,1] using 2 swaps (using the circular property of the array). [1,1,1,1,1,0,0,0,0] using 2 swaps. There is no way to group all 1's together with 0 or 1 swaps. Thus, the minimum number of swaps required is 2.

Example 3: Input: nums = [1,1,0,0,1] Output: 0 Explanation: All the 1's are already grouped together due to the circular property of the array. Thus, the minimum number of swaps required is 0.

Constraints:

1 <= nums.length <= 105 nums[i] is either 0 or 1.

==
Problem Number: 1407 URL: https://leetcode.com/problems/count-words-obtained-after-adding-a-letter Title: 2135. Count Words Obtained After Adding a Letter Problem Description: You are given two 0-indexed arrays of strings startWords and targetWords. Each string consists of lowercase English letters only. For each string in targetWords, check if it is possible to choose a string from startWords and perform a conversion operation on it to be equal to that from targetWords. The conversion operation is described in the following two steps:

Append any lowercase letter that is not present in the string to its end.

For example, if the string is "abc", the letters 'd', 'e', or 'y' can be added to it, but not 'a'. If 'd' is added, the resulting string will be "abcd".

Rearrange the letters of the new string in any arbitrary order.

For example, "abcd" can be rearranged to "acbd", "bacd", "cbda", and so on. Note that it can also be rearranged to "abcd" itself.

Return the number of strings in targetWords that can be obtained by performing the operations on any string of startWords. Note that you will only be verifying if the string in targetWords can be obtained from a string in startWords by performing the operations. The strings in startWords do not actually change during this process. Example 1: Input: startWords = ["ant","act","tack"], targetWords = ["tack","act","acti"] Output: 2 Explanation: - In order to form targetWords[0] = "tack", we use startWords[1] = "act", append 'k' to it, and rearrange "actk" to "tack". - There is no string in startWords that can be used to obtain targetWords[1] = "act". Note that "act" does exist in startWords, but we must append one letter to the string before rearranging it. - In order to form targetWords[2] = "acti", we use startWords[1] = "act", append 'i' to it, and rearrange "acti" to "acti" itself.

Example 2: Input: startWords = ["ab","a"], targetWords = ["abc","abcd"] Output: 1 Explanation: - In order to form targetWords[0] = "abc", we use startWords[0] = "ab", add 'c' to it, and rearrange it to "abc". - There is no string in startWords that can be used to obtain targetWords[1] = "abcd".

Constraints:

1 <= startWords.length, targetWords.length <= 5 * 104 1 <= startWords[i].length, targetWords[j].length <= 26 Each string of startWords and targetWords consists of lowercase English letters only. No letter occurs more than once in any string of startWords or targetWords.

==

Problem Number: 1408 URL: https://leetcode.com/problems/minimum-moves-to-reach-target-score Title: 2139. Minimum Moves to Reach Target Score Problem Description: You are playing a game with integers. You start with the integer 1 and you want to reach the integer target. In one move, you can either:

Increment the current integer by one (i.e., x = x + 1). Double the current integer (i.e., x = 2 * x).

You can use the increment operation any number of times, however, you can only use the double operation at most maxDoubles times. Given the two integers target and maxDoubles, return the minimum number of moves needed to reach target starting with 1. Example 1: Input: target = 5, maxDoubles = 0 Output: 4 Explanation: Keep incrementing by 1 until you reach target.

Example 2: Input: target = 19, maxDoubles = 2 Output: 7 Explanation: Initially, x = 1 Increment 3 times so x = 4 Double once so x = 8 Increment once so x = 9 Double again so x = 18 Increment once so x = 19

Example 3: Input: target = 10, maxDoubles = 4 Output: 4 Explanation: Initially, x = 1 Increment once so x = 2 Double once so x = 4 Increment once so x = 5 Double again so x = 10

Constraints:

1 <= target <= 109 0 <= maxDoubles <= 100

==

Problem Number: 1409 URL: https://leetcode.com/problems/solving-questions-with-brainpower Title: 2140. Solving Questions With Brainpower Problem Description: You are given a 0-indexed 2D integer array questions where questions[i] = [pointsi, brainpoweri]. The array describes the questions of an exam, where you have to process the questions in order (i.e., starting from question 0) and make a decision whether to solve or skip each question. Solving question i will earn you pointsi points but you will be unable to solve each of the next brainpoweri questions. If you skip question i, you get to make the decision on the next question.

For example, given questions = [[3, 2], [4, 3], [4, 4], [2, 5]]:

If question 0 is solved, you will earn 3 points but you will be unable to solve questions 1 and 2. If instead, question 0 is skipped and question 1 is solved, you will earn 4 points but you will be unable to solve questions 2 and 3.

Return the maximum points you can earn for the exam. Example 1: Input: questions = [[3,2],[4,3],[4,4],[2,5]] Output: 5 Explanation: The maximum points can be earned by solving questions 0 and 3. - Solve question 0: Earn 3 points, will be unable to solve the next 2 questions - Unable to solve questions 1 and 2 - Solve question 3: Earn 2 points Total points earned: 3 + 2 = 5. There is no other way to earn 5 or more points.

Example 2: Input: questions = [[1,1],[2,2],[3,3],[4,4],[5,5]] Output: 7 Explanation: The maximum points can be earned by solving questions 1 and 4. - Skip question 0 - Solve question 1: Earn 2 points, will be unable to solve the next 2 questions - Unable to solve questions 2 and 3 - Solve question 4: Earn 5 points Total points earned: 2 + 5 = 7. There is no other way to earn 7 or more points.

Constraints:

1 <= questions.length <= 105 questions[i].length == 2 1 <= pointsi, brainpoweri <= 105

==

Problem Number: 1410 URL: https://leetcode.com/problems/count-the-hidden-sequences Title: 2145. Count the Hidden Sequences Problem Description: You are given a 0-indexed array of n integers differences, which describes the differences between each pair of consecutive integers of a hidden sequence of length (n + 1). More formally, call the hidden sequence hidden, then we have that differences[i] = hidden[i + 1] - hidden[i]. You are further given two integers lower and upper that describe the inclusive range of values [lower, upper] that the hidden sequence can contain.

For example, given differences = [1, -3, 4], lower = 1, upper = 6, the hidden sequence is a sequence of length 4 whose elements are in between 1 and 6 (inclusive).

[3, 4, 1, 5] and [4, 5, 2, 6] are possible hidden sequences. [5, 6, 3, 7] is not possible since it contains an element greater than 6. [1, 2, 3, 4] is not possible since the differences are not correct.

Return the number of possible hidden sequences there are. If there are no possible sequences, return 0.   Example 1: Input: differences = [1,-3,4], lower = 1, upper = 6 Output: 2 Explanation: The possible hidden sequences are: - [3, 4, 1, 5] - [4, 5, 2, 6] Thus, we return 2.

Example 2: Input: differences = [3,-4,5,1,-2], lower = -4, upper = 5 Output: 4 Explanation: The possible hidden sequences are: - [-3, 0, -4, 1, 2, 0] - [-2, 1, -3, 2, 3, 1] - [-1, 2, -2, 3, 4, 2] - [0, 3, -1, 4, 5, 3] Thus, we return 4.

Example 3: Input: differences = [4,-7,2], lower = 3, upper = 6 Output: 0 Explanation: There are no possible hidden sequences. Thus, we return 0.

Constraints:

n == differences.length 1 <= n <= 105 -105 <= differences[i] <= 105 -105 <= lower <= upper <= 105

==
Problem Number: 1411 URL: https://leetcode.com/problems/k-highest-ranked-items-within-a-price-range Title: 2146. K Highest Ranked Items Within a Price Range Problem Description: You are given a 0-indexed 2D integer array grid of size m x n that represents a map of the items in a shop. The integers in the grid represent the following:

0 represents a wall that you cannot pass through. 1 represents an empty cell that you can freely move to and from. All other positive integers represent the price of an item in that cell. You may also freely move to and from these item cells.

It takes 1 step to travel between adjacent grid cells. You are also given integer arrays pricing and start where pricing = [low, high] and start = [row, col] indicates that you start at the position (row, col) and are interested only in items with a price in the range of [low, high] (inclusive). You are further given an integer k. You are interested in the positions of the k highest-ranked items whose prices are within the given price range. The rank is determined by the first of these criteria that is different:

Distance, defined as the length of the shortest path from the start (shorter distance has a higher rank). Price (lower price has a higher rank, but it must be in the price range). The row number (smaller row number has a higher rank). The column number (smaller column number has a higher rank).

Return the k highest-ranked items within the price range sorted by their rank (highest to lowest). If there are fewer than k reachable items within the price range, return all of them.   Example 1:

Input: grid = [[1,2,0,1],[1,3,0,1],[0,2,5,1]], pricing = [2,5], start = [0,0], k = 3 Output: [[0,1],[1,1],[2,1]] Explanation: You start at (0,0). With a price range of [2,5], we can take items from (0,1), (1,1), (2,1) and (2,2). The ranks of these items are: - (0,1) with distance 1 - (1,1) with distance 2 - (2,1) with distance 3 - (2,2) with distance 4 Thus, the 3 highest ranked items in the price range are (0,1), (1,1), and (2,1).

Example 2:

Input: grid = [[1,2,0,1],[1,3,3,1],[0,2,5,1]], pricing = [2,3], start = [2,3], k = 2 Output: [[2,1],[1,2]] Explanation: You start at (2,3). With a price range of [2,3], we can take items from (0,1), (1,1), (1,2) and (2,1). The ranks of these items are: - (2,1) with distance 2, price 2 - (1,2) with distance 2, price 3 - (1,1) with distance 3 - (0,1) with distance 4 Thus, the 2 highest ranked items in the price range are (2,1) and (1,2).

Example 3:

Input: grid = [[1,1,1],[0,0,1],[2,3,4]], pricing = [2,3], start = [0,0], k = 3 Output: [[2,1],[2,0]] Explanation: You start at (0,0). With a price range of [2,3], we can take items from (2,0) and (2,1). The ranks of these items are: - (2,1) with distance 5 - (2,0) with distance 6 Thus, the 2 highest ranked items in the price range are (2,1) and (2,0). Note that k = 3 but there are only 2 reachable items within the price range.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 1 <= m * n <= 105 0 <= grid[i][j] <= 105 pricing.length == 2 2 <= low <= high <= 105 start.length == 2 0 <= row <= m - 1 0 <= col <= n - 1 grid[row][col] > 0 1 <= k <= m * n

==
Problem Number: 1412 URL: https://leetcode.com/problems/rearrange-array-elements-by-sign Title: 2149. Rearrange Array Elements by Sign Problem Description: You are given a 0-indexed integer array nums of even length consisting of an equal number of positive and negative integers. You should rearrange the elements of nums such that the modified array follows the given conditions:

Every consecutive pair of integers have opposite signs. For all integers with the same sign, the order in which they were present in nums is preserved. The rearranged array begins with a positive integer.

Return the modified array after rearranging the elements to satisfy the aforementioned conditions. Example 1: Input: nums = [3,1,-2,-5,2,-4] Output: [3,-2,1,-5,2,-4] Explanation: The positive integers in nums are [3,1,2]. The negative integers are [-2,-5,-4]. The only possible way to rearrange them such that they satisfy all conditions is [3,-2,1,-5,2,-4]. Other ways such as [1,-2,2,-5,3,-4], [3,1,2,-2,-5,-4], [-2,3,-5,1,-4,2] are incorrect because they do not satisfy one or more conditions.

Example 2: Input: nums = [-1,1] Output: [1,-1] Explanation: 1 is the only positive integer and -1 the only negative integer in nums. So nums is rearranged to [1,-1].

Constraints:

2 <= nums.length <= 2 * 105 nums.length is even 1 <= |nums[i]| <= 105 nums consists of equal number of positive and negative integers.

==
Problem Number: 1413 URL: https://leetcode.com/problems/find-all-lonely-numbers-in-the-array Title: 2150. Find All Lonely Numbers in the Array Problem Description: You are given an integer array nums. A number x is lonely when it appears only once, and no adjacent numbers (i.e. x + 1 and x - 1) appear in the array. Return all lonely numbers in nums. You may return the answer in any order. Example 1: Input: nums = [10,6,5,8] Output: [10,8] Explanation: - 10 is a lonely number since it appears exactly once and 9 and 11 does not appear in nums. - 8 is a lonely number since it appears exactly once and 7 and 9 does not appear in nums. - 5 is not a lonely number since 6 appears in nums and vice versa. Hence, the lonely numbers in nums are [10, 8]. Note that [8, 10] may also be returned.

Example 2: Input: nums = [1,3,5,3] Output: [1,5] Explanation: - 1 is a lonely number since it appears exactly once and 0 and 2 does not appear in nums. - 5 is a lonely number since it appears exactly once and 4 and 6 does not appear in nums. - 3 is not a lonely number since it appears twice. Hence, the lonely numbers in nums are [1, 5]. Note that [5, 1] may also be returned.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 106

==
Problem Number: 1414 URL: https://leetcode.com/problems/all-divisions-with-the-highest-score-of-a-binary-array Title: 2155. All Divisions With the Highest Score of a Binary Array Problem Description: You are given a 0-indexed binary array nums of length n. nums can be divided at index i (where 0 <= i <= n) into two arrays (possibly empty) numsleft and numsright:

numsleft has all the elements of nums between index 0 and i - 1 (inclusive), while numsright has all the elements of nums between index i and n - 1 (inclusive). If i == 0, numsleft is empty, while numsright has all the elements of nums. If i == n, numsleft has all the elements of nums, while numsright is empty.

The division score of an index i is the sum of the number of 0's in numsleft and the number of 1's in numsright. Return all distinct indices that have the highest possible division score. You may return the answer in any order. Example 1: Input: nums = [0,0,1,0] Output: [2,4] Explanation: Division at index - 0: numsleft is []. numsright is [0,0,1,0]. The score is 0 + 1 = 1. - 1: numsleft is [0]. numsright is [0,1,0]. The score is 1 + 1 = 2. - 2: numsleft is [0,0]. numsright is [1,0]. The score is 2 + 1 = 3. - 3: numsleft is [0,0,1]. numsright is [0]. The score is 2 + 0 = 2. - 4: numsleft is [0,0,1,0]. numsright is []. The score is 3 + 0

= 3. Indices 2 and 4 both have the highest possible division score 3. Note the answer [4,2] would also be accepted. Example 2: Input: nums = [0,0,0] Output: [3] Explanation: Division at index - 0: numsleft is []. numsright is [0,0,0]. The score is 0 + 0 = 0. - 1: numsleft is [0]. numsright is [0,0]. The score is 1 + 0 = 1. - 2: numsleft is [0,0]. numsright is [0]. The score is 2 + 0 = 2. - 3: numsleft is [0,0,0]. numsright is []. The score is 3 + 0 = 3. Only index 3 has the highest possible division score 3.

Example 3: Input: nums = [1,1] Output: [0] Explanation: Division at index - 0: numsleft is []. numsright is [1,1]. The score is 0 + 2 = 2. - 1: numsleft is [1]. numsright is [1]. The score is 0 + 1 = 1. - 2: numsleft is [1,1]. numsright is []. The score is 0 + 0 = 0. Only index 0 has the highest possible division score 2.

Constraints:

n == nums.length 1 <= n <= 105 nums[i] is either 0 or 1.

==
Problem Number: 1415 URL: https://leetcode.com/problems/partition-array-according-to-given-pivot Title: 2161. Partition Array According to Given Pivot Problem Description: You are given a 0-indexed integer array nums and an integer pivot. Rearrange nums such that the following conditions are satisfied:

Every element less than pivot appears before every element greater than pivot. Every element equal to pivot appears in between the elements less than and greater than pivot. The relative order of the elements less than pivot and the elements greater than pivot is maintained.

More formally, consider every pi, pj where pi is the new position of the ith element and pj is the new position of the jth element. For elements less than pivot, if i < j and nums[i] < pivot and nums[j] < pivot, then pi < pj. Similarly for elements greater than pivot, if i < j and nums[i] > pivot and nums[j] > pivot, then pi < pj.

Return nums after the rearrangement. Example 1: Input: nums = [9,12,5,10,14,3,10], pivot = 10 Output: [9,5,3,10,10,12,14] Explanation: The elements 9, 5, and 3 are less than the pivot so they are on the left side of the array. The elements 12 and 14 are greater than the pivot so they are on the right side of the array. The relative ordering of the elements less than and greater than pivot is also maintained. [9, 5, 3] and [12, 14] are the respective orderings.

Example 2: Input: nums = [-3,4,3,2], pivot = 2 Output: [-3,2,4,3] Explanation: The element -3 is less than the pivot so it is on the left side of the array. The elements 4 and 3 are greater than the pivot so they are on the right side of the array. The relative ordering of the elements less than and greater than pivot is also maintained. [-3] and [4, 3] are the respective orderings.

Constraints:

1 <= nums.length <= 105 -106 <= nums[i] <= 106 pivot equals to an element of nums.

== Problem Number: 1416 URL: https://leetcode.com/problems/minimum-cost-to-set-cooking-time Title: 2162. Minimum Cost to Set Cooking Time Problem Description: A generic microwave supports cooking times for:

at least 1 second. at most 99 minutes and 99 seconds.

To set the cooking time, you push at most four digits. The microwave normalizes what you push as four digits by prepending zeroes. It interprets the first two digits as the minutes and the last two digits as the seconds. It then adds them up as the cooking time. For example,

You push 9 5 4 (three digits). It is normalized as 0954 and interpreted as 9 minutes and 54 seconds. You push 0 0 0 8 (four digits). It is interpreted as 0 minutes and 8 seconds. You push 8 0 9 0. It is interpreted as 80 minutes and 90 seconds. You push 8 1 3 0. It is interpreted as 81 minutes and 30 seconds.

You are given integers startAt, moveCost, pushCost, and targetSeconds. Initially, your finger is on the digit startAt. Moving the finger above any specific digit costs moveCost units of fatigue. Pushing the digit below the finger once costs pushCost units of fatigue. There can be multiple ways to set the microwave to cook for targetSeconds seconds but you are interested in the way with the minimum cost. Return the minimum cost to set targetSeconds seconds of cooking time. Remember that one minute consists of 60 seconds.   Example 1:

Input: startAt = 1, moveCost = 2, pushCost = 1, targetSeconds = 600 Output: 6 Explanation: The following are the possible ways to set the cooking time. - 1 0 0 0, interpreted as 10 minutes and 0 seconds.   The finger is already on digit 1, pushes 1 (with cost 1), moves to 0 (with cost 2), pushes 0 (with cost 1), pushes 0 (with cost 1), and pushes 0 (with cost 1).   The cost is: 1 + 2 + 1 + 1 + 1 = 6. This is the minimum cost. - 0 9 6 0, interpreted as 9 minutes and 60 seconds. That is also 600 seconds.   The finger moves to 0 (with cost 2), pushes 0 (with cost 1), moves to 9 (with cost 2), pushes 9 (with cost 1), moves to 6 (with cost 2), pushes 6 (with cost 1), moves to 0 (with cost 2), and pushes 0 (with cost 1).   The cost is: 2 + 1 + 2 + 1 + 2 + 1 + 2 + 1 = 12. - 9 6 0, normalized as 0960 and interpreted as 9 minutes and 60 seconds.   The finger moves to 9 (with cost 2), pushes 9 (with cost 1), moves to 6 (with cost 2), pushes 6 (with cost 1), moves to 0 (with cost 2), and pushes 0 (with cost 1). The cost is: 2 + 1 + 2 + 1 + 2 + 1 = 9.

Example 2:

Input: startAt = 0, moveCost = 1, pushCost = 2, targetSeconds = 76 Output: 6 Explanation: The optimal way is to push two digits: 7 6, interpreted as 76 seconds. The finger moves to 7 (with cost 1), pushes 7 (with cost 2), moves to 6 (with cost 1), and pushes 6 (with cost 2). The total cost is: 1 + 2 + 1 + 2

= 6 Note other possible ways are 0076, 076, 0116, and 116, but none of them produces the minimum cost.

Constraints:

0 <= startAt <= 9 1 <= moveCost, pushCost <= 105 1 <= targetSeconds <= 6039

==

Problem Number: 1417 URL: https://leetcode.com/problems/smallest-value-of-the-rearranged-number Title: 2165. Smallest Value of the Rearranged Number Problem Description: You are given an integer num. Rearrange the digits of num such that its value is minimized and it does not contain any leading zeros. Return the rearranged number with minimal value. Note that the sign of the number does not change after rearranging the digits. Example 1: Input: num = 310 Output: 103 Explanation: The possible arrangements for the digits of 310 are 013, 031, 103, 130, 301, 310. The arrangement with the smallest value that does not contain any leading zeros is 103.

Example 2: Input: num = -7605 Output: -7650 Explanation: Some possible arrangements for the digits of -7605 are -7650, -6705, -5076, -0567. The arrangement with the smallest value that does not contain any leading zeros is -7650.

Constraints:

-1015 <= num <= 1015

==

Problem Number: 1418 URL: https://leetcode.com/problems/design-bitset Title: 2166. Design Bitset Problem Description: A Bitset is a data structure that compactly stores bits. Implement the Bitset class:

Bitset(int size) Initializes the Bitset with size bits, all of which are 0. void fix(int idx) Updates the value of the bit at the index idx to 1. If the value was already 1, no change occurs. void unfix(int idx) Updates the value of the bit at the index idx to 0. If the value was already 0, no change occurs. void flip() Flips the values of each bit in the Bitset. In other words, all bits with value 0 will now have value 1 and vice versa. boolean all() Checks if the value of each bit in the Bitset is 1. Returns true if it satisfies the condition, false otherwise. boolean one() Checks if there is at least one bit in the Bitset with value 1. Returns true if it satisfies the condition, false otherwise. int count() Returns the total number of bits in the Bitset which have value 1. String toString() Returns the current composition of the Bitset. Note that in the resultant string, the character at the ith index should coincide with the value at the ith bit of the Bitset.

Example 1: Input ["Bitset", "fix", "fix", "flip", "all", "unfix", "flip", "one", "unfix", "count", "toString"] [[5], [3], [1], [], [], [0], [], [], [0], [], []] Output [null, null, null, null, false, null, null, true, null, 2, "01010"]

Explanation Bitset bs = new Bitset(5); // bitset = "00000". bs.fix(3); // the value at idx = 3 is updated to 1, so bitset = "00010". bs.fix(1); // the value at idx = 1 is updated to 1, so bitset = "01010". bs.flip(); // the value of each bit is flipped, so bitset = "10101". bs.all(); // return False, as not all values of the bitset are 1. bs.unfix(0); // the value at idx = 0 is updated to 0, so bitset = "00101". bs.flip(); // the value of each bit is flipped, so bitset = "11010". bs.one(); // return True, as there is at least 1 index with value 1. bs.unfix(0); // the value at idx = 0 is updated to 0, so bitset = "01010". bs.count(); // return 2, as there are 2 bits with value 1. bs.toString(); // return "01010", which is the composition of bitset.

Constraints:

1 <= size <= 105 0 <= idx <= size - 1 At most 105 calls will be made in total to fix, unfix, flip, all, one, count, and toString. At least one call will be made to all, one, count, or toString. At most 5 calls will be made to toString.

==
Problem Number: 1419 URL: https://leetcode.com/problems/minimum-operations-to-make-the-array-alternating Title: 2170. Minimum Operations to Make the Array Alternating Problem Description: You are given a 0-indexed array nums consisting of n positive integers. The array nums is called alternating if:

nums[i - 2] == nums[i], where 2 <= i <= n - 1. nums[i - 1] != nums[i], where 1 <= i <= n - 1.

In one operation, you can choose an index i and change nums[i] into any positive integer. Return the minimum number of operations required to make the array alternating. Example 1: Input: nums = [3,1,3,2,4,3] Output: 3 Explanation: One way to make the array alternating is by converting it to [3,1,3,1,3,1]. The number of operations required in this case is 3. It can be proven that it is not possible to make the array alternating in less than 3 operations.

Example 2: Input: nums = [1,2,2,2,2] Output: 2 Explanation: One way to make the array alternating is by converting it to [1,2,1,2,1]. The number of operations required in this case is 2. Note that the array cannot be converted to [2,2,2,2,2] because in this case nums[0] == nums[1] which violates the conditions of an alternating array.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105

==
Problem Number: 1420 URL: https://leetcode.com/problems/removing-minimum-number-of-magic-beans Title: 2171. Removing Minimum Number of Magic Beans Problem Description: You are given an array of positive integers beans, where each integer represents the number of magic beans found in a particular magic bag. Remove any number of beans (possibly none) from each

bag such that the number of beans in each remaining non-empty bag (still containing at least one bean) is equal. Once a bean has been removed from a bag, you are not allowed to return it to any of the bags. Return the minimum number of magic beans that you have to remove.   Example 1: Input: beans = [4,1,6,5] Output: 4 Explanation: - We remove 1 bean from the bag with only 1 bean. This results in the remaining bags: [4,0,6,5] - Then we remove 2 beans from the bag with 6 beans. This results in the remaining bags: [4,0,4,5] - Then we remove 1 bean from the bag with 5 beans. This results in the remaining bags: [4,0,4,4] We removed a total of 1 + 2 + 1 = 4 beans to make the remaining non-empty bags have an equal number of beans. There are no other solutions that remove 4 beans or fewer.

Example 2: Input: beans = [2,10,3,2] Output: 7 Explanation: - We remove 2 beans from one of the bags with 2 beans. This results in the remaining bags: [0,10,3,2] - Then we remove 2 beans from the other bag with 2 beans. This results in the remaining bags: [0,10,3,0] - Then we remove 3 beans from the bag with 3 beans. This results in the remaining bags: [0,10,0,0] We removed a total of 2 + 2 + 3 = 7 beans to make the remaining non-empty bags have an equal number of beans. There are no other solutions that removes 7 beans or fewer.

Constraints:

1 <= beans.length <= 105 1 <= beans[i] <= 105

== Problem Number: 1421 URL: https://leetcode.com/problems/find-three-consecutive-integers-that-sum-to-a-given-number Title: 2177. Find Three Consecutive Integers That Sum to a Given Number Problem Description: Given an integer num, return three consecutive integers (as a sorted array) that sum to num. If num cannot be expressed as the sum of three consecutive integers, return an empty array.   Example 1: Input: num = 33 Output: [10,11,12] Explanation: 33 can be expressed as 10 + 11 + 12 = 33. 10, 11, 12 are 3 consecutive integers, so we return [10, 11, 12].

Example 2: Input: num = 4 Output: [] Explanation: There is no way to express 4 as the sum of 3 consecutive integers.

Constraints:

0 <= num <= 1015

== Problem Number: 1422 URL: https://leetcode.com/problems/maximum-split-of-positive-even-integers Title: 2178. Maximum Split of Positive Even Integers Problem Description: You are given an integer finalSum. Split it into a sum of a maximum number of unique positive even integers.

For example, given finalSum = 12, the following splits are valid (unique positive even integers summing up to finalSum): (12), (2 + 10), (2 + 4 + 6), and (4 + 8). Among them, (2 + 4 + 6) contains the maximum number of integers. Note that finalSum cannot be split into (2 + 2 + 4 + 4) as all the numbers should be unique.

Return a list of integers that represent a valid split containing a maximum number of integers. If no valid split exists for finalSum, return an empty list. You may return the integers in any order.   Example 1: Input: finalSum = 12 Output: [2,4,6] Explanation: The following are valid splits: (12), (2 + 10), (2 + 4 + 6), and (4 + 8). (2 + 4 + 6) has the maximum number of integers, which is 3. Thus, we return [2,4,6]. Note that [2,6,4], [6,2,4], etc. are also accepted.

Example 2: Input: finalSum = 7 Output: [] Explanation: There are no valid splits for the given finalSum. Thus, we return an empty array.

Example 3: Input: finalSum = 28 Output: [6,8,2,12] Explanation: The following are valid splits: (2 + 26), (6 + 8 + 2 + 12), and (4 + 24). (6 + 8 + 2 + 12) has the maximum number of integers, which is 4. Thus, we return [6,8,2,12]. Note that [10,2,4,12], [6,2,4,16], etc. are also accepted.

Constraints:

1 <= finalSum <= 1010

== Problem Number: 1423 URL: https://leetcode.com/problems/merge-nodes-in-between-zeros Title: 2181. Merge Nodes in Between Zeros Problem Description: You are given the head of a linked list, which contains a series of integers separated by 0's. The beginning and end of the linked list will have Node.val == 0. For every two consecutive 0's, merge all the nodes lying in between them into a single node whose value is the sum of all the merged nodes. The modified list should not contain any 0's. Return the head of the modified linked list.   Example 1:

Input: head = [0,3,1,0,4,5,2,0] Output: [4,11] Explanation: The above figure represents the given linked list. The modified list contains - The sum of the nodes marked in green: 3 + 1 = 4. - The sum of the nodes marked in red: 4 + 5 + 2 = 11.

Example 2:

Input: head = [0,1,0,3,0,2,2,0] Output: [1,3,4] Explanation: The above figure represents the given linked list. The modified list contains - The sum of the nodes marked in green: 1 = 1. - The sum of the nodes marked in red: 3 = 3. - The sum of the nodes marked in yellow: 2 + 2 = 4.

Constraints:

The number of nodes in the list is in the range [3, 2 * 105]. 0 <= Node.val <= 1000 There are no two consecutive nodes with Node.val == 0. The beginning and end of the linked list have Node.val == 0.

== Problem Number: 1424 URL: https://leetcode.com/problems/construct-string-with-repeat-limit Title: 2182. Construct String With Repeat Limit Problem Description: You are given a string s and an integer repeatLimit. Construct a new string repeatLimitedString using the characters of s such that no letter appears more than repeatLimit times in a row. You do not have to use all characters from s. Return the lexicographically largest repeatLimitedString possible. A string a is lexicographically larger than a string b if in the first position where a and b differ, string a has a letter that appears later in the alphabet than the corresponding letter in b. If the first min(a.length, b.length) characters do not differ, then the longer string is the lexicographically larger one. Example 1: Input: s = "cczazcc", repeatLimit = 3 Output: "zzcccac" Explanation: We use all of the characters from s to construct the repeatLimitedString "zzcccac". The letter 'a' appears at most 1 time in a row. The letter 'c' appears at most 3 times in a row. The letter 'z' appears at most 2 times in a row. Hence, no letter appears more than repeatLimit times in a row and the string is a valid repeatLimitedString. The string is the lexicographically largest repeatLimitedString possible so we return "zzcccac". Note that the string "zzcccca" is lexicographically larger but the letter 'c' appears more than 3 times in a row, so it is not a valid repeatLimitedString.

Example 2: Input: s = "aababab", repeatLimit = 2 Output: "bbabaa" Explanation: We use only some of the characters from s to construct the repeatLimitedString "bbabaa". The letter 'a' appears at most 2 times in a row. The letter 'b' appears at most 2 times in a row. Hence, no letter appears more than repeatLimit times in a row and the string is a valid repeatLimitedString. The string is the lexicographically largest repeatLimitedString possible so we return "bbabaa". Note that the string "bbabaaa" is lexicographically larger but the letter 'a' appears more than 2 times in a row, so it is not a valid repeatLimitedString.

Constraints:

1 <= repeatLimit <= s.length <= 105 s consists of lowercase English letters.

==
Problem Number: 1425 URL: https://leetcode.com/problems/minimum-number-of-steps-to-make-two-strings-anagram-ii Title: 2186. Minimum Number of Steps to Make Two Strings Anagram II Problem Description: You are given two strings s and t. In one step, you can append any character to either s or t. Return the minimum number of steps to make s and t anagrams of each other. An anagram of a string is a string that contains the same characters with a different (or the same) ordering. Example 1: Input: s = "leetcode", t = "coats" Output: 7 Explanation: - In 2 steps, we can append the letters in "as" onto s = "leetcode", forming s = "leetcodeas". - In 5 steps, we can append the letters in "leede" onto t = "coats", forming t = "coatsleede". "leetcodeas" and "coatsleede" are now anagrams of each other. We used a total of 2 + 5 = 7 steps. It can be shown that there is no way to make them anagrams of each other with less than 7 steps.

Example 2: Input: s = ”night”, t = ”thing” Output: 0 Explanation: The given strings are already anagrams of each other. Thus, we do not need any further steps.

Constraints:

1 <= s.length, t.length <= 2 * 105 s and t consist of lowercase English letters.

==
Problem Number: 1426 URL: https://leetcode.com/problems/minimum-time-to-complete-trips Title: 2187. Minimum Time to Complete Trips Problem Description: You are given an array time where time[i] denotes the time taken by the ith bus to complete one trip. Each bus can make multiple trips successively; that is, the next trip can start immediately after completing the current trip. Also, each bus operates independently; that is, the trips of one bus do not influence the trips of any other bus. You are also given an integer totalTrips, which denotes the number of trips all buses should make in total. Return the minimum time required for all buses to complete at least totalTrips trips. Example 1: Input: time = [1,2,3], totalTrips = 5 Output: 3 Explanation: - At time t = 1, the number of trips completed by each bus are [1,0,0]. The total number of trips completed is 1 + 0 + 0 = 1. - At time t = 2, the number of trips completed by each bus are [2,1,0]. The total number of trips completed is 2 + 1 + 0 = 3. - At time t = 3, the number of trips completed by each bus are [3,1,1]. The total number of trips completed is 3 + 1 + 1 = 5. So the minimum time needed for all buses to complete at least 5 trips is 3.

Example 2: Input: time = [2], totalTrips = 1 Output: 2 Explanation: There is only one bus, and it will complete its first trip at t = 2. So the minimum time needed to complete 1 trip is 2.

Constraints:

1 <= time.length <= 105 1 <= time[i], totalTrips <= 107

==
Problem Number: 1427 URL: https://leetcode.com/problems/sort-the-jumbled-numbers Title: 2191. Sort the Jumbled Numbers Problem Description: You are given a 0-indexed integer array mapping which represents the mapping rule of a shuffled decimal system. mapping[i] = j means digit i should be mapped to digit j in this system. The mapped value of an integer is the new integer obtained by replacing each occurrence of digit i in the integer with mapping[i] for all 0 <= i <= 9. You are also given another integer array nums. Return the array nums sorted in non-decreasing order based on the mapped values of its elements. Notes:

Elements with the same mapped values should appear in the same relative order as in the input. The elements of nums should only be sorted based on their mapped values and not be replaced by them.

Example 1: Input: mapping = [8,9,4,0,2,1,3,5,7,6], nums = [991,338,38] Output: [338,38,991] Explanation: Map the number 991 as follows: 1. mapping[9] = 6, so all occurrences of the digit 9 will become 6. 2. mapping[1] = 9, so all occurrences of the digit 1 will become 9. Therefore, the mapped value of 991 is 669. 338 maps to 007, or 7 after removing the leading zeros. 38 maps to 07, which is also 7 after removing leading zeros. Since 338 and 38 share the same mapped value, they should remain in the same relative order, so 338 comes before 38. Thus, the sorted array is [338,38,991].

Example 2: Input: mapping = [0,1,2,3,4,5,6,7,8,9], nums = [789,456,123] Output: [123,456,789] Explanation: 789 maps to 789, 456 maps to 456, and 123 maps to 123. Thus, the sorted array is [123,456,789].

Constraints:

mapping.length == 10 0 <= mapping[i] <= 9 All the values of mapping[i] are unique. 1 <= nums.length <= 3 * 104 0 <= nums[i] < 109

==
Problem Number: 1428 URL: https://leetcode.com/problems/all-ancestors-of-a-node-in-a-directed-acyclic-graph Title: 2192. All Ancestors of a Node in a Directed Acyclic Graph Problem Description: You are given a positive integer n representing the number of nodes of a Directed Acyclic Graph (DAG). The nodes are numbered from 0 to n - 1 (inclusive). You are also given a 2D integer array edges, where edges[i] = [fromi, toi] denotes that there is a unidirectional edge from fromi to toi in the graph. Return a list answer, where answer[i] is the list of ancestors of the ith node, sorted in ascending order. A node u is an ancestor of another node v if u can reach v via a set of edges. Example 1:

Input: n = 8, edgeList = [[0,3],[0,4],[1,3],[2,4],[2,7],[3,5],[3,6],[3,7],[4,6]] Output: [[],[],[],[0,1],[0,2],[0,1,3],[0,1,2,3,4],[0,1,2,3]] Explanation: The above diagram represents the input graph. - Nodes 0, 1, and 2 do not have any ancestors. - Node 3 has two ancestors 0 and 1. - Node 4 has two ancestors 0 and 2. - Node 5 has three ancestors 0, 1, and 3. - Node 6 has five ancestors 0, 1, 2, 3, and 4. - Node 7 has four ancestors 0, 1, 2, and 3.

Example 2:

Input: n = 5, edgeList = [[0,1],[0,2],[0,3],[0,4],[1,2],[1,3],[1,4],[2,3],[2,4],[3,4]] Output: [[],[0],[0,1],[0,1,2],[0,1,2,3]] Explanation: The above diagram represents the input graph. - Node 0 does not have any ancestor. - Node 1 has one ancestor 0. - Node 2 has two ancestors 0 and 1. - Node 3 has three ancestors 0, 1, and 2. - Node 4 has four ancestors 0, 1, 2, and 3.

Constraints:

1 <= n <= 1000 0 <= edges.length <= min(2000, n * (n - 1) / 2) edges[i].length == 2 0 <= fromi, toi <= n - 1 fromi != toi There are no duplicate edges. The graph is directed and acyclic.

==
Problem Number: 1429 URL: https://leetcode.com/problems/append-k-integers-with-minimal-sum Title: 2195. Append K Integers With Minimal Sum Problem Description: You are given an integer array nums and an integer k. Append k unique positive integers that do not appear in nums to nums such that the resulting total sum is minimum. Return the sum of the k integers appended to nums. Example 1: Input: nums = [1,4,25,10,25], k = 2 Output: 5 Explanation: The two unique positive integers that do not appear in nums which we append are 2 and 3. The resulting sum of nums is 1 + 4 + 25 + 10 + 25 + 2 + 3 = 70, which is the minimum. The sum of the two integers appended is 2 + 3 = 5, so we return 5. Example 2: Input: nums = [5,6], k = 6 Output: 25 Explanation: The six unique positive integers that do not appear in nums which we append are 1, 2, 3, 4, 7, and 8. The resulting sum of nums is 5 + 6 + 1 + 2 + 3 + 4 + 7 + 8 = 36, which is the minimum. The sum of the six integers appended is 1 + 2 + 3 + 4 + 7 + 8 = 25, so we return 25.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= k <= 108

==
Problem Number: 1430 URL: https://leetcode.com/problems/create-binary-tree-from-descriptions Title: 2196. Create Binary Tree From Descriptions Problem Description: You are given a 2D integer array descriptions where descriptions[i] = [parenti, childi, isLefti] indicates that parenti is the parent of childi in a binary tree of unique values. Furthermore,

If isLefti == 1, then childi is the left child of parenti. If isLefti == 0, then childi is the right child of parenti.

Construct the binary tree described by descriptions and return its root. The test cases will be generated such that the binary tree is valid. Example 1:

Input: descriptions = [[20,15,1],[20,17,0],[50,20,1],[50,80,0],[80,19,1]] Output: [50,20,80,15,17,19] Explanation: The root node is the node with value 50 since it has no parent. The resulting binary tree is shown in the diagram.

Example 2:

Input: descriptions = [[1,2,1],[2,3,0],[3,4,1]] Output: [1,2,null,null,3,4] Explanation: The root node is the node with value 1 since it has no parent. The resulting binary tree is shown in the diagram.

Constraints:

1 <= descriptions.length <= 104 descriptions[i].length == 3 1 <= parenti, childi <= 105 0 <= isLefti <= 1 The binary tree described by descriptions is valid.

==
Problem Number: 1431 URL: https://leetcode.com/problems/count-artifacts-

that-can-be-extracted Title: 2201. Count Artifacts That Can Be Extracted Problem Description: There is an n x n 0-indexed grid with some artifacts buried in it. You are given the integer n and a 0-indexed 2D integer array artifacts describing the positions of the rectangular artifacts where artifacts[i] = [r1i, c1i, r2i, c2i] denotes that the ith artifact is buried in the subgrid where:

(r1i, c1i) is the coordinate of the top-left cell of the ith artifact and (r2i, c2i) is the coordinate of the bottom-right cell of the ith artifact.

You will excavate some cells of the grid and remove all the mud from them. If the cell has a part of an artifact buried underneath, it will be uncovered. If all the parts of an artifact are uncovered, you can extract it. Given a 0-indexed 2D integer array dig where dig[i] = [ri, ci] indicates that you will excavate the cell (ri, ci), return the number of artifacts that you can extract. The test cases are generated such that:

No two artifacts overlap. Each artifact only covers at most 4 cells. The entries of dig are unique.

Example 1:

Input: n = 2, artifacts = [[0,0,0,0],[0,1,1,1]], dig = [[0,0],[0,1]] Output: 1 Explanation: The different colors represent different artifacts. Excavated cells are labeled with a 'D' in the grid. There is 1 artifact that can be extracted, namely the red artifact. The blue artifact has one part in cell (1,1) which remains uncovered, so we cannot extract it. Thus, we return 1.

Example 2:

Input: n = 2, artifacts = [[0,0,0,0],[0,1,1,1]], dig = [[0,0],[0,1],[1,1]] Output: 2 Explanation: Both the red and blue artifacts have all parts uncovered (labeled with a 'D') and can be extracted, so we return 2.

Constraints:

1 <= n <= 1000 1 <= artifacts.length, dig.length <= min(n2, 105) artifacts[i].length == 4 dig[i].length == 2 0 <= r1i, c1i, r2i, c2i, ri, ci <= n - 1 r1i <= r2i c1i <= c2i No two artifacts will overlap. The number of cells covered by an artifact is at most 4. The entries of dig are unique.

==
Problem Number: 1432 URL: https://leetcode.com/problems/maximize-the-topmost-element-after-k-moves Title: 2202. Maximize the Topmost Element After K Moves Problem Description: You are given a 0-indexed integer array nums representing the contents of a pile, where nums[0] is the topmost element of the pile. In one move, you can perform either of the following:

If the pile is not empty, remove the topmost element of the pile. If there are one or more removed elements, add any one of them back onto the pile. This element becomes the new topmost element.

You are also given an integer k, which denotes the total number of moves to be made. Return the maximum value of the topmost element of the pile possible after exactly k moves. In case it is not possible to obtain a non-empty pile after k moves, return -1. Example 1: Input: nums = [5,2,2,4,0,6], k = 4 Output: 5 Explanation: One of the ways we can end with 5 at the top of the pile after 4 moves is as follows: - Step 1: Remove the topmost element = 5. The pile becomes [2,2,4,0,6]. - Step 2: Remove the topmost element = 2. The pile becomes [2,4,0,6]. - Step 3: Remove the topmost element = 2. The pile becomes [4,0,6]. - Step 4: Add 5 back onto the pile. The pile becomes [5,4,0,6]. Note that this is not the only way to end with 5 at the top of the pile. It can be shown that 5 is the largest answer possible after 4 moves.

Example 2: Input: nums = [2], k = 1 Output: -1 Explanation: In the first move, our only option is to pop the topmost element of the pile. Since it is not possible to obtain a non-empty pile after one move, we return -1.

Constraints:

1 <= nums.length <= 105 0 <= nums[i], k <= 109

==

Problem Number: 1433 URL: https://leetcode.com/problems/maximize-number-of-subsequences-in-a-string Title: 2207. Maximize Number of Subsequences in a String Problem Description: You are given a 0-indexed string text and another 0-indexed string pattern of length 2, both of which consist of only lowercase English letters. You can add either pattern[0] or pattern[1] anywhere in text exactly once. Note that the character can be added even at the beginning or at the end of text. Return the maximum number of times pattern can occur as a subsequence of the modified text. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. Example 1: Input: text = "abdcdbc", pattern = "ac" Output: 4 Explanation: If we add pattern[0] = 'a' in between text[1] and text[2], we get "abadcdbc". Now, the number of times "ac" occurs as a subsequence is 4. Some other strings which have 4 subsequences "ac" after adding a character to text are "aabdcdbc" and "abdacdbc". However, strings such as "abdcadbc", "abdccdbc", and "abdcdbcc", although obtainable, have only 3 subsequences "ac" and are thus suboptimal. It can be shown that it is not possible to get more than 4 subsequences "ac" by adding only one character.

Example 2: Input: text = "aabb", pattern = "ab" Output: 6 Explanation: Some of the strings which can be obtained from text and have 6 subsequences "ab" are "aaabb", "aaabb", and "aabbb".

Constraints:

1 <= text.length <= 105 pattern.length == 2 text and pattern consist only of lowercase English letters.

==

Problem Number: 1434 URL: https://leetcode.com/problems/minimum-operations-to-halve-array-sum Title: 2208. Minimum Operations to Halve Array Sum Problem Description: You are given an array nums of positive integers. In one operation, you can choose any number from nums and reduce it to exactly half the number. (Note that you may choose this reduced number in future operations.) Return the minimum number of operations to reduce the sum of nums by at least half. Example 1: Input: nums = [5,19,8,1] Output: 3 Explanation: The initial sum of nums is equal to 5 + 19 + 8 + 1 = 33. The following is one of the ways to reduce the sum by at least half: Pick the number 19 and reduce it to 9.5. Pick the number 9.5 and reduce it to 4.75. Pick the number 8 and reduce it to 4. The final array is [5, 4.75, 4, 1] with a total sum of 5 + 4.75 + 4 + 1 = 14.75. The sum of nums has been reduced by 33 - 14.75 = 18.25, which is at least half of the initial sum, 18.25 >= 33/2 = 16.5. Overall, 3 operations were used so we return 3. It can be shown that we cannot reduce the sum by at least half in less than 3 operations.

Example 2: Input: nums = [3,8,20] Output: 3 Explanation: The initial sum of nums is equal to 3 + 8 + 20 = 31. The following is one of the ways to reduce the sum by at least half: Pick the number 20 and reduce it to 10. Pick the number 10 and reduce it to 5. Pick the number 3 and reduce it to 1.5. The final array is [1.5, 8, 5] with a total sum of 1.5 + 8 + 5 = 14.5. The sum of nums has been reduced by 31 - 14.5 = 16.5, which is at least half of the initial sum, 16.5 >= 31/2 = 15.5. Overall, 3 operations were used so we return 3. It can be shown that we cannot reduce the sum by at least half in less than 3 operations.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 107

==

Problem Number: 1435 URL: https://leetcode.com/problems/count-collisions-on-a-road Title: 2211. Count Collisions on a Road Problem Description: There are n cars on an infinitely long road. The cars are numbered from 0 to n - 1 from left to right and each car is present at a unique point. You are given a 0-indexed string directions of length n. directions[i] can be either 'L', 'R', or 'S' denoting whether the ith car is moving towards the left, towards the right, or staying at its current point respectively. Each moving car has the same speed. The number of collisions can be calculated as follows:

When two cars moving in opposite directions collide with each other, the number of collisions increases by 2. When a moving car collides with a stationary car, the number of collisions increases by 1.

After a collision, the cars involved can no longer move and will stay at the point where they collided. Other than that, cars cannot change their state or direction of motion. Return the total number of collisions that will happen on the road. Example 1: Input: directions = "RLRSLL" Output: 5 Explanation: The collisions that will happen on the road are: - Cars 0 and 1 will collide with each other. Since they are moving in opposite directions, the number of collisions becomes 0 + 2 = 2. - Cars 2 and 3 will collide with each other. Since car 3 is stationary, the number of collisions becomes 2 + 1 = 3. - Cars 3 and 4 will collide with each other. Since car 3 is stationary, the number of collisions becomes 3 + 1 = 4. - Cars 4 and 5 will collide with each other. After car 4 collides with car 3, it will stay at the point of collision and get hit by car 5. The number of collisions becomes 4 + 1 = 5. Thus, the total number of collisions that will happen on the road is 5.

Example 2: Input: directions = "LLRR" Output: 0 Explanation: No cars will collide with each other. Thus, the total number of collisions that will happen on the road is 0.   Constraints:

1 <= directions.length <= 105 directions[i] is either 'L', 'R', or 'S'.

== Problem Number: 1436 URL: https://leetcode.com/problems/maximum-points-in-an-archery-competition Title: 2212. Maximum Points in an Archery Competition Problem Description: Alice and Bob are opponents in an archery competition. The competition has set the following rules:

Alice first shoots numArrows arrows and then Bob shoots numArrows arrows. The points are then calculated as follows:

The target has integer scoring sections ranging from 0 to 11 inclusive. For each section of the target with score k (in between 0 to 11), say Alice and Bob have shot ak and bk arrows on that section respectively. If ak >= bk, then Alice takes k points. If ak < bk, then Bob takes k points. However, if ak == bk == 0, then nobody takes k points.

For example, if Alice and Bob both shot 2 arrows on the section with score 11, then Alice takes 11 points. On the other hand, if Alice shot 0 arrows on the section with score 11 and Bob shot 2 arrows on that same section, then Bob takes 11 points.

You are given the integer numArrows and an integer array aliceArrows of size 12, which represents the number of arrows Alice shot on each scoring section from 0 to 11. Now, Bob wants to maximize the total number of points he can obtain. Return the array bobArrows which represents the number of arrows Bob shot on each scoring section from 0 to 11. The sum of the values in bobArrows should equal numArrows. If there are multiple ways for Bob to earn the maximum total points, return any one of them.   Example 1:

Input: numArrows = 9, aliceArrows = [1,1,0,1,0,0,2,1,0,1,2,0] Output: [0,0,0,0,1,1,0,0,1,2,3,1] Explanation: The table above shows how the competition is scored. Bob earns a total point of 4 + 5 + 8 + 9 + 10 + 11 = 47. It can be shown that Bob cannot obtain a score higher than 47 points.

Example 2:

Input: numArrows = 3, aliceArrows = [0,0,1,0,0,0,0,0,0,0,0,2] Output: [0,0,0,0,0,0,0,0,1,1,1,0] Explanation: The table above shows how the competition is scored. Bob earns a total point of 8 + 9 + 10 = 27. It can be shown that Bob cannot obtain a score higher than 27 points.

Constraints:

1 <= numArrows <= 105 aliceArrows.length == bobArrows.length == 12 0 <= aliceArrows[i], bobArrows[i] <= numArrows sum(aliceArrows[i]) == numArrows

==
Problem Number: 1437 URL: https://leetcode.com/problems/minimum-deletions-to-make-array-beautiful Title: 2216. Minimum Deletions to Make Array Beautiful Problem Description: You are given a 0-indexed integer array nums. The array nums is beautiful if:

nums.length is even. nums[i] != nums[i + 1] for all i % 2 == 0.

Note that an empty array is considered beautiful. You can delete any number of elements from nums. When you delete an element, all the elements to the right of the deleted element will be shifted one unit to the left to fill the gap created and all the elements to the left of the deleted element will remain unchanged. Return the minimum number of elements to delete from nums to make it beautiful. Example 1: Input: nums = [1,1,2,3,5] Output: 1 Explanation: You can delete either nums[0] or nums[1] to make nums = [1,2,3,5] which is beautiful. It can be proven you need at least 1 deletion to make nums beautiful.

Example 2: Input: nums = [1,1,2,2,3,3] Output: 2 Explanation: You can delete nums[0] and nums[5] to make nums = [1,2,2,3] which is beautiful. It can be proven you need at least 2 deletions to make nums beautiful.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 105

==
Problem Number: 1438 URL: https://leetcode.com/problems/find-palindrome-with-fixed-length Title: 2217. Find Palindrome With Fixed Length Problem Description: Given an integer array queries and a positive integer intLength, return an array answer where answer[i] is either the queries[i]th smallest positive palindrome of length intLength or -1 if no such palindrome exists. A palindrome is a number that reads the same backwards and forwards. Palindromes cannot have leading zeros. Example 1: Input: queries = [1,2,3,4,5,90], intLength = 3 Output: [101,111,121,131,141,999] Explanation: The first few palindromes of length 3 are: 101, 111, 121, 131, 141, 151, 161, 171, 181, 191, 202, ... The 90th palindrome of length 3 is 999.

Example 2: Input: queries = [2,4,6], intLength = 4 Output: [1111,1331,1551] Explanation: The first six palindromes of length 4 are: 1001, 1111, 1221, 1331, 1441, and 1551.

Constraints:

1 <= queries.length <= 5 * 104 1 <= queries[i] <= 109 1 <= intLength <= 15

==
Problem Number: 1439 URL: https://leetcode.com/problems/find-triangular-sum-of-an-array Title: 2221. Find Triangular Sum of an Array Problem Description: You are given a 0-indexed integer array nums, where nums[i] is a digit between 0 and 9 (inclusive). The triangular sum of nums is the value of the only element present in nums after the following process terminates:

Let nums comprise of n elements. If n == 1, end the process. Otherwise, create a new 0-indexed integer array newNums of length n - 1. For each index i, where 0 <= i < n - 1, assign the value of newNums[i] as (nums[i] + nums[i+1]) % 10, where % denotes modulo operator. Replace the array nums with newNums. Repeat the entire process starting from step 1.

Return the triangular sum of nums. Example 1:

Input: nums = [1,2,3,4,5] Output: 8 Explanation: The above diagram depicts the process from which we obtain the triangular sum of the array. Example 2: Input: nums = [5] Output: 5 Explanation: Since there is only one element in nums, the triangular sum is the value of that element itself. Constraints:

1 <= nums.length <= 1000 0 <= nums[i] <= 9

==
Problem Number: 1440 URL: https://leetcode.com/problems/number-of-ways-to-select-buildings Title: 2222. Number of Ways to Select Buildings Problem Description: You are given a 0-indexed binary string s which represents the types of buildings along a street where:

s[i] = '0' denotes that the ith building is an office and s[i] = '1' denotes that the ith building is a restaurant.

As a city official, you would like to select 3 buildings for random inspection. However, to ensure variety, no two consecutive buildings out of the selected buildings can be of the same type.

For example, given s = "001101", we cannot select the 1st, 3rd, and 5th buildings as that would form "011" which is not allowed due to having two consecutive buildings of the same type.

Return the number of valid ways to select 3 buildings. Example 1: Input: s = "001101" Output: 6 Explanation: The following sets of indices selected are valid: - [0,2,4] from "001101" forms "010" - [0,3,4] from "001101" forms "010" - [1,2,4] from "001101" forms "010" - [1,3,4] from "001101" forms "010" - [2,4,5] from "001101" forms "101" - [3,4,5] from "001101" forms "101" No other selection is valid. Thus, there are 6 total ways.

Example 2: Input: s = "11100" Output: 0 Explanation: It can be shown that there are no valid selections.

Constraints:

3 <= s.length <= 105 s[i] is either '0' or '1'.

==
Problem Number: 1441 URL: https://leetcode.com/problems/find-players-with-zero-or-one-losses Title: 2225. Find Players With Zero or One Losses Problem Description: You are given an integer array matches where matches[i] = [winneri, loseri] indicates that the player winneri defeated player loseri in a match. Return a list answer of size 2 where:

answer[0] is a list of all players that have not lost any matches. answer[1] is a list of all players that have lost exactly one match.

The values in the two lists should be returned in increasing order. Note:

You should only consider the players that have played at least one match. The testcases will be generated such that no two matches will have the same outcome.

Example 1: Input: matches = [[1,3],[2,3],[3,6],[5,6],[5,7],[4,5],[4,8],[4,9],[10,4],[10,9]] Output: [[1,2,10],[4,5,7,8]] Explanation: Players 1, 2, and 10 have not lost any matches. Players 4, 5, 7, and 8 each have lost one match. Players 3, 6, and 9 each have lost two matches. Thus, answer[0] = [1,2,10] and answer[1] = [4,5,7,8].

Example 2: Input: matches = [[2,3],[1,3],[5,4],[6,4]] Output: [[1,2,5,6],[]] Explanation: Players 1, 2, 5, and 6 have not lost any matches. Players 3 and 4 each have lost two matches. Thus, answer[0] = [1,2,5,6] and answer[1] = [].

Constraints:

1 <= matches.length <= 105 matches[i].length == 2 1 <= winneri, loseri <= 105 winneri != loseri All matches[i] are unique.

==
Problem Number: 1442 URL: https://leetcode.com/problems/maximum-candies-allocated-to-k-children Title: 2226. Maximum Candies Allocated to K Children Problem Description: You are given a 0-indexed integer array candies. Each element in the array denotes a pile of candies of size candies[i]. You can divide each pile into any number of sub piles, but you cannot merge two piles together. You are also given an integer k. You should allocate piles of candies to k children such that each child gets the same number of candies. Each child can take at most one pile of candies and some piles of candies may go unused. Return the maximum number of candies each child can get. Example 1: Input: candies = [5,8,6], k = 3 Output: 5 Explanation: We can divide candies[1] into 2 piles of size 5 and 3, and candies[2] into 2 piles of size 5 and 1. We now have five piles of candies of sizes 5, 5, 3, 5, and 1. We can allocate the 3 piles of size 5 to 3 children. It can be proven that each child cannot receive more than 5 candies.

Example 2: Input: candies = [2,5], k = 11 Output: 0 Explanation: There are

11 children but only 7 candies in total, so it is impossible to ensure each child receives at least one candy. Thus, each child gets no candy and the answer is 0.

Constraints:

1 <= candies.length <= 105 1 <= candies[i] <= 107 1 <= k <= 1012

==

Problem Number: 1443 URL: https://leetcode.com/problems/minimize-result-by-adding-parentheses-to-expression Title: 2232. Minimize Result by Adding Parentheses to Expression Problem Description: You are given a 0-indexed string expression of the form "<num1>+<num2>" where <num1> and <num2> represent positive integers. Add a pair of parentheses to expression such that after the addition of parentheses, expression is a valid mathematical expression and evaluates to the smallest possible value. The left parenthesis must be added to the left of '+' and the right parenthesis must be added to the right of '+'. Return expression after adding a pair of parentheses such that expression evaluates to the smallest possible value. If there are multiple answers that yield the same result, return any of them. The input has been generated such that the original value of expression, and the value of expression after adding any pair of parentheses that meets the requirements fits within a signed 32-bit integer. Example 1: Input: expression = "247+38" Output: "2(47+38)" Explanation: The expression evaluates to 2 * (47 + 38) = 2 * 85 = 170. Note that "2(4)7+38" is invalid because the right parenthesis must be to the right of the '+'. It can be shown that 170 is the smallest possible value.

Example 2: Input: expression = "12+34" Output: "1(2+3)4" Explanation: The expression evaluates to 1 * (2 + 3) * 4 = 1 * 5 * 4 = 20.

Example 3: Input: expression = "999+999" Output: "(999+999)" Explanation: The expression evaluates to 999 + 999 = 1998.

Constraints:

3 <= expression.length <= 10 expression consists of digits from '1' to '9' and '+'. expression starts and ends with digits. expression contains exactly one '+'. The original value of expression, and the value of expression after adding any pair of parentheses that meets the requirements fits within a signed 32-bit integer.

==

Problem Number: 1444 URL: https://leetcode.com/problems/maximum-product-after-k-increments Title: 2233. Maximum Product After K Increments Problem Description: You are given an array of non-negative integers nums and an integer k. In one operation, you may choose any element from nums and increment it by 1. Return the maximum product of nums after at most k operations. Since the answer may be very large, return it modulo 109 + 7. Note that you should maximize the product before taking the modulo. Example 1: Input: nums = [0,4], k = 5 Output: 20 Explanation: Increment the first number 5 times. Now nums = [5, 4], with a product of 5 * 4 = 20. It can be shown that 20 is maximum product possible, so we return 20. Note that there may be other ways to increment nums to have the maximum product.

Example 2: Input: nums = [6,3,3,2], k = 2 Output: 216 Explanation: Increment the second number 1 time and increment the fourth number 1 time. Now nums = [6, 4, 3, 3], with a product of 6 * 4 * 3 * 3 = 216. It can be shown that 216 is maximum product possible, so we return 216. Note that there may be other ways to increment nums to have the maximum product.

Constraints:

1 <= nums.length, k <= 105 0 <= nums[i] <= 106

==
Problem Number: 1445 URL: https://leetcode.com/problems/number-of-ways-to-buy-pens-and-pencils Title: 2240. Number of Ways to Buy Pens and Pencils Problem Description: You are given an integer total indicating the amount of money you have. You are also given two integers cost1 and cost2 indicating the price of a pen and pencil respectively. You can spend part or all of your money to buy multiple quantities (or none) of each kind of writing utensil. Return the number of distinct ways you can buy some number of pens and pencils. Example 1: Input: total = 20, cost1 = 10, cost2 = 5 Output: 9 Explanation: The price of a pen is 10 and the price of a pencil is 5. - If you buy 0 pens, you can buy 0, 1, 2, 3, or 4 pencils. - If you buy 1 pen, you can buy 0, 1, or 2 pencils. - If you buy 2 pens, you cannot buy any pencils. The total number of ways to buy pens and pencils is 5 + 3 + 1 = 9.

Example 2: Input: total = 5, cost1 = 10, cost2 = 10 Output: 1 Explanation: The price of both pens and pencils are 10, which cost more than total, so you cannot buy any writing utensils. Therefore, there is only 1 way: buy 0 pens and 0 pencils.

Constraints:

1 <= total, cost1, cost2 <= 106

==
Problem Number: 1446 URL: https://leetcode.com/problems/design-an-atm-machine Title: 2241. Design an ATM Machine Problem Description: There is an ATM machine that stores banknotes of 5 denominations: 20, 50, 100, 200, and 500 dollars. Initially the ATM is empty. The user can use the machine to deposit or withdraw any amount of money. When withdrawing, the machine prioritizes using banknotes of larger values.

For example, if you want to withdraw $300 and there are 2 $50 banknotes, 1 $100 banknote, and 1 $200 banknote, then the machine will use the $100 and $200 banknotes. However, if you try to withdraw $600 and there are 3 $200 banknotes and 1 $500 banknote, then the withdraw request will be rejected because the machine will first try to use the $500 banknote and then be unable to use banknotes to complete the remaining $100. Note that the machine is not allowed to use the $200 banknotes instead of the $500 banknote.

Implement the ATM class:

ATM() Initializes the ATM object. void deposit(int[] banknotesCount) Deposits new banknotes in the order $20, $50, $100, $200, and $500. int[] withdraw(int amount) Returns an array of length 5 of the number of banknotes that will be handed to the user in the order $20, $50, $100, $200, and $500, and update the number of banknotes in the ATM after withdrawing. Returns [-1] if it is not possible (do not withdraw any banknotes in this case).

Example 1: Input ["ATM", "deposit", "withdraw", "deposit", "withdraw", "withdraw"] [[], [[0,0,1,2,1]], [600], [[0,1,0,1,1]], [600], [550]] Output [null, null, [0,0,1,0,1], null, [-1], [0,1,0,0,1]]

Explanation ATM atm = new ATM(); atm.deposit([0,0,1,2,1]); // Deposits 1 $100 banknote, 2 $200 banknotes, // and 1 $500 banknote. atm.withdraw(600); // Returns [0,0,1,0,1]. The machine uses 1 $100 banknote // and 1 $500 banknote. The banknotes left over in the // machine are [0,0,0,2,0]. atm.deposit([0,1,0,1,1]); // Deposits 1 $50, $200, and $500 banknote. // The banknotes in the machine are now [0,1,0,3,1]. atm.withdraw(600); // Returns [-1]. The machine will try to use a $500 banknote // and then be unable to complete the remaining $100, // so the withdraw request will be rejected. // Since the request is rejected, the number of banknotes // in the machine is not modified. atm.withdraw(550); // Returns [0,1,0,0,1]. The machine uses 1 $50 banknote // and 1 $500 banknote. Constraints:

banknotesCount.length == 5 0 <= banknotesCount[i] <= 109 1 <= amount <= 109 At most 5000 calls in total will be made to withdraw and deposit. At least one call will be made to each function withdraw and deposit.

==
Problem Number: 1447 URL: https://leetcode.com/problems/minimum-rounds-to-complete-all-tasks Title: 2244. Minimum Rounds to Complete All Tasks Problem Description: You are given a 0-indexed integer array tasks, where tasks[i] represents the difficulty level of a task. In each round, you can complete either 2 or 3 tasks of the same difficulty level. Return the minimum rounds required to complete all the tasks, or -1 if it is not possible to complete all the tasks. Example 1: Input: tasks = [2,2,3,3,2,4,4,4,4,4] Output: 4 Explanation: To complete all the tasks, a possible plan is: - In the first round, you complete 3 tasks of difficulty level 2. - In the second round, you complete 2 tasks of difficulty level 3. - In the third round, you complete 3 tasks of difficulty level 4. - In the fourth round, you complete 2 tasks of difficulty level 4. It can be shown that all the tasks cannot be completed in fewer than 4 rounds, so the answer is 4.

Example 2: Input: tasks = [2,3,3] Output: -1 Explanation: There is only 1 task of difficulty level 2, but in each round, you can only complete either 2 or 3 tasks of the same difficulty level. Hence, you cannot complete all the tasks, and the answer is -1.

Constraints:

1 <= tasks.length <= 105 1 <= tasks[i] <= 109

==
Problem Number: 1448 URL: https://leetcode.com/problems/maximum-trailing-zeros-in-a-cornered-path Title: 2245. Maximum Trailing Zeros in a Cornered Path Problem Description: You are given a 2D integer array grid of size m x n, where each cell contains a positive integer. A cornered path is defined as a set of adjacent cells with at most one turn. More specifically, the path should exclusively move either horizontally or vertically up to the turn (if there is one), without returning to a previously visited cell. After the turn, the path will then move exclusively in the alternate direction: move vertically if it moved horizontally, and vice versa, also without returning to a previously visited cell. The product of a path is defined as the product of all the values in the path. Return the maximum number of trailing zeros in the product of a cornered path found in grid. Note:

Horizontal movement means moving in either the left or right direction. Vertical movement means moving in either the up or down direction.

Example 1:

Input: grid = [[23,17,15,3,20],[8,1,20,27,11],[9,4,6,2,21],[40,9,1,10,6],[22,7,4,5,3]] Output: 3 Explanation: The grid on the left shows a valid cornered path. It has a product of 15 * 20 * 6 * 1 * 10 = 18000 which has 3 trailing zeros. It can be shown that this is the maximum trailing zeros in the product of a cornered path.

The grid in the middle is not a cornered path as it has more than one turn. The grid on the right is not a cornered path as it requires a return to a previously visited cell.

Example 2:

Input: grid = [[4,3,2],[7,6,1],[8,8,8]] Output: 0 Explanation: The grid is shown in the figure above. There are no cornered paths in the grid that result in a product with a trailing zero.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 1 <= m * n <= 105 1 <= grid[i][j] <= 1000

==
Problem Number: 1449 URL: https://leetcode.com/problems/count-lattice-points-inside-a-circle Title: 2249. Count Lattice Points Inside a Circle Problem Description: Given a 2D integer array circles where circles[i] = [xi, yi, ri] represents the center (xi, yi) and radius ri of the ith circle drawn on a grid,

return the number of lattice points that are present inside at least one circle. Note:

A lattice point is a point with integer coordinates. Points that lie on the circumference of a circle are also considered to be inside it.

Example 1:

Input: circles = [[2,2,1]] Output: 5 Explanation: The figure above shows the given circle. The lattice points present inside the circle are (1, 2), (2, 1), (2, 2), (2, 3), and (3, 2) and are shown in green. Other points such as (1, 1) and (1, 3), which are shown in red, are not considered inside the circle. Hence, the number of lattice points present inside at least one circle is 5. Example 2:

Input: circles = [[2,2,2],[3,4,1]] Output: 16 Explanation: The figure above shows the given circles. There are exactly 16 lattice points which are present inside at least one circle. Some of them are (0, 2), (2, 0), (2, 4), (3, 2), and (4, 4).

Constraints:

1 <= circles.length <= 200 circles[i].length == 3 1 <= xi, yi <= 100 1 <= ri <= min(xi, yi)

==
Problem Number: 1450 URL: https://leetcode.com/problems/count-number-of-rectangles-containing-each-point Title: 2250. Count Number of Rectangles Containing Each Point Problem Description: You are given a 2D integer array rectangles where rectangles[i] = [li, hi] indicates that ith rectangle has a length of li and a height of hi. You are also given a 2D integer array points where points[j] = [xj, yj] is a point with coordinates (xj, yj). The ith rectangle has its bottom-left corner point at the coordinates (0, 0) and its top-right corner point at (li, hi). Return an integer array count of length points.length where count[j] is the number of rectangles that contain the jth point. The ith rectangle contains the jth point if 0 <= xj <= li and 0 <= yj <= hi. Note that points that lie on the edges of a rectangle are also considered to be contained by that rectangle. Example 1:

Input: rectangles = [[1,2],[2,3],[2,5]], points = [[2,1],[1,4]] Output: [2,1] Explanation: The first rectangle contains no points. The second rectangle contains only the point (2, 1). The third rectangle contains the points (2, 1) and (1, 4). The number of rectangles that contain the point (2, 1) is 2. The number of rectangles that contain the point (1, 4) is 1. Therefore, we return [2, 1].

Example 2:

Input: rectangles = [[1,1],[2,2],[3,3]], points = [[1,3],[1,1]] Output: [1,3] Explanation: The first rectangle contains only the point (1, 1). The second rectangle contains only the point (1, 1). The third rectangle contains the points (1, 3) and (1, 1). The number of rectangles that contain the point (1, 3) is 1. The number of rectangles that contain the point (1, 1) is 3. Therefore, we return [1, 3].

Constraints:

1 <= rectangles.length, points.length <= 5 * 104 rectangles[i].length == points[j].length == 2 1 <= li, xj <= 109 1 <= hi, yj <= 100 All the rectangles are unique. All the points are unique.

==
Problem Number: 1451 URL: https://leetcode.com/problems/minimum-average-difference Title: 2256. Minimum Average Difference Problem Description: You are given a 0-indexed integer array nums of length n. The average difference of the index i is the absolute difference between the average of the first i + 1 elements of nums and the average of the last n - i - 1 elements. Both averages should be rounded down to the nearest integer. Return the index with the minimum average difference. If there are multiple such indices, return the smallest one. Note:

The absolute difference of two numbers is the absolute value of their difference. The average of n elements is the sum of the n elements divided (integer division) by n. The average of 0 elements is considered to be 0.

Example 1: Input: nums = [2,5,3,9,5,3] Output: 3 Explanation: - The average difference of index 0 is: |2 / 1 - (5 + 3 + 9 + 5 + 3) / 5| = |2 / 1 - 25 / 5| = |2 - 5| = 3. - The average difference of index 1 is: |(2 + 5) / 2 - (3 + 9 + 5 + 3) / 4| = |7 / 2 - 20 / 4| = |3 - 5| = 2. - The average difference of index 2 is: |(2 + 5 + 3) / 3 - (9 + 5 + 3) / 3| = |10 / 3 - 17 / 3| = |3 - 5| = 2. - The average difference of index 3 is: |(2 + 5 + 3 + 9) / 4 - (5 + 3) / 2| = |19 / 4 - 8 / 2| = |4 - 4| = 0. - The average difference of index 4 is: |(2 + 5 + 3 + 9 + 5) / 5 - 3 / 1| = |24 / 5 - 3 / 1| = |4 - 3| = 1. - The average difference of index 5 is: |(2 + 5 + 3 + 9 + 5 + 3) / 6 - 0| = |27 / 6 - 0| = |4 - 0| = 4. The average difference of index 3 is the minimum average difference so return 3.

Example 2: Input: nums = [0] Output: 0 Explanation: The only index is 0 so return 0. The average difference of index 0 is: |0 / 1 - 0| = |0 - 0| = 0.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 105

==
Problem Number: 1452 URL: https://leetcode.com/problems/count-unguarded-cells-in-the-grid Title: 2257. Count Unguarded Cells in the Grid Problem Description: You are given two integers m and n representing a 0-indexed m x n grid. You are also given two 2D integer arrays guards and walls where guards[i] = [rowi, coli] and walls[j] = [rowj, colj] represent the positions of the ith guard and jth wall respectively. A guard can see every cell in the four cardinal directions (north, east, south, or west) starting from their position unless obstructed by a wall or another guard. A cell is guarded if there is at least one guard that can see it. Return the number of unoccupied cells that are not guarded. Example 1:

Input: m = 4, n = 6, guards = [[0,0],[1,1],[2,3]], walls = [[0,1],[2,2],[1,4]] Output: 7 Explanation: The guarded and unguarded cells are shown in red and green respectively in the above diagram. There are a total of 7 unguarded cells, so we return 7.

Example 2:

Input: m = 3, n = 3, guards = [[1,1]], walls = [[0,1],[1,0],[2,1],[1,2]] Output: 4 Explanation: The unguarded cells are shown in green in the above diagram. There are a total of 4 unguarded cells, so we return 4.

Constraints:

1 <= m, n <= 105 2 <= m * n <= 105 1 <= guards.length, walls.length <= 5 * 104 2 <= guards.length + walls.length <= m * n guards[i].length == walls[j].length == 2 0 <= rowi, rowj < m 0 <= coli, colj < n All the positions in guards and walls are unique.

==
Problem Number: 1453 URL: https://leetcode.com/problems/minimum-consecutive-cards-to-pick-up Title: 2260. Minimum Consecutive Cards to Pick Up Problem Description: You are given an integer array cards where cards[i] represents the value of the ith card. A pair of cards are matching if the cards have the same value. Return the minimum number of consecutive cards you have to pick up to have a pair of matching cards among the picked cards. If it is impossible to have matching cards, return -1. Example 1: Input: cards = [3,4,2,3,4,7] Output: 4 Explanation: We can pick up the cards [3,4,2,3] which contain a matching pair of cards with value 3. Note that picking up the cards [4,2,3,4] is also optimal.

Example 2: Input: cards = [1,0,5,3] Output: -1 Explanation: There is no way to pick up a set of consecutive cards that contain a pair of matching cards.

Constraints:

1 <= cards.length <= 105 0 <= cards[i] <= 106

==
Problem Number: 1454 URL: https://leetcode.com/problems/k-divisible-elements-subarrays Title: 2261. K Divisible Elements Subarrays Problem Description: Given an integer array nums and two integers k and p, return the number of distinct subarrays, which have at most k elements that are divisible by p. Two arrays nums1 and nums2 are said to be distinct if:

They are of different lengths, or There exists at least one index i where nums1[i] != nums2[i].

A subarray is defined as a non-empty contiguous sequence of elements in an array. Example 1: Input: nums = [2,3,3,2,2], k = 2, p = 2 Output: 11 Explanation: The elements at indices 0, 3, and 4 are divisible by p = 2. The 11 distinct subarrays which have at most k = 2 elements divisible by 2 are: [2], [2,3], [2,3,3], [2,3,3,2], [3], [3,3], [3,3,2], [3,3,2,2], [3,2], [3,2,2], and [2,2]. Note that the subarrays [2] and [3] occur more than once in nums, but they should each be counted only once. The subarray [2,3,3,2,2] should not be counted because it has 3 elements that are divisible by 2.

Example 2: Input: nums = [1,2,3,4], k = 4, p = 1 Output: 10 Explanation: All element of nums are divisible by p = 1. Also, every subarray of nums will have at most 4 elements that are divisible by 1. Since all subarrays are distinct, the total number of subarrays satisfying all the constraints is 10.

Constraints:

1 <= nums.length <= 200 1 <= nums[i], p <= 200 1 <= k <= nums.length

Follow up: Can you solve this problem in O(n2) time complexity?

==
Problem Number: 1455 URL: https://leetcode.com/problems/count-nodes-equal-to-average-of-subtree Title: 2265. Count Nodes Equal to Average of Subtree Problem Description: Given the root of a binary tree, return the number of nodes where the value of the node is equal to the average of the values in its subtree. Note:

The average of n elements is the sum of the n elements divided by n and rounded down to the nearest integer. A subtree of root is a tree consisting of root and all of its descendants.

Example 1:

Input: root = [4,8,5,0,1,null,6] Output: 5 Explanation: For the node with value 4: The average of its subtree is (4 + 8 + 5 + 0 + 1 + 6) / 6 = 24 / 6 = 4. For the node with value 5: The average of its subtree is (5 + 6) / 2 = 11 / 2 = 5. For the node with value 0: The average of its subtree is 0 / 1 = 0. For the node with value 1: The average of its subtree is 1 / 1 = 1. For the node with value 6: The average of its subtree is 6 / 1 = 6.

Example 2:

Input: root = [1] Output: 1 Explanation: For the node with value 1: The average of its subtree is 1 / 1 = 1.

Constraints:

The number of nodes in the tree is in the range [1, 1000]. 0 <= Node.val <= 1000

==
Problem Number: 1456 URL: https://leetcode.com/problems/count-number-of-texts Title: 2266. Count Number of Texts Problem Description: Alice is texting Bob using her phone. The mapping of digits to letters is shown in the figure below.

In order to add a letter, Alice has to press the key of the corresponding digit i times, where i is the position of the letter in the key.

For example, to add the letter 's', Alice has to press '7' four times. Similarly, to add the letter 'k', Alice has to press '5' twice. Note that the digits '0' and '1' do not map to any letters, so Alice does not use them.

However, due to an error in transmission, Bob did not receive Alice's text message but received a string of pressed keys instead.

For example, when Alice sent the message "bob", Bob received the string "2266622".

Given a string pressedKeys representing the string received by Bob, return the total number of possible text messages Alice could have sent. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: pressedKeys = "22233" Output: 8 Explanation: The possible text messages Alice could have sent are: "aaadd", "abdd", "badd", "cdd", "aaae", "abe", "bae", and "ce". Since there are 8 possible messages, we return 8.

Example 2: Input: pressedKeys = "222222222222222222222222222222222222" Output: 82876089 Explanation: There are 2082876103 possible text messages Alice could have sent. Since we need to return the answer modulo 109 + 7, we return 2082876103 % (109 + 7) = 82876089.

Constraints:

1 <= pressedKeys.length <= 105 pressedKeys only consists of digits from '2' - '9'.

==

Problem Number: 1457 URL: https://leetcode.com/problems/number-of-ways-to-split-array Title: 2270. Number of Ways to Split Array Problem Description: You are given a 0-indexed integer array nums of length n. nums contains a valid split at index i if the following are true:

The sum of the first i + 1 elements is greater than or equal to the sum of the last n - i - 1 elements. There is at least one element to the right of i. That is, 0 <= i < n - 1.

Return the number of valid splits in nums. Example 1: Input: nums = [10,4,-8,7] Output: 2 Explanation: There are three ways of splitting nums into two non-empty parts: - Split nums at index 0. Then, the first part is [10], and its sum is 10. The second part is [4,-8,7], and its sum is 3. Since 10 >= 3, i = 0 is a valid split. - Split nums at index 1. Then, the first part is [10,4], and its sum is 14. The second part is [-8,7], and its sum is -1. Since 14 >= -1, i = 1 is a valid split. - Split nums at index 2. Then, the first part is [10,4,-8], and its sum is 6. The second part is [7], and its sum is 7. Since 6 < 7, i = 2 is not a valid split. Thus, the number of valid splits in nums is 2.

Example 2: Input: nums = [2,3,1,0] Output: 2 Explanation: There are two valid splits in nums: - Split nums at index 1. Then, the first part is [2,3], and its sum is 5. The second part is [1,0], and its sum is 1. Since 5 >= 1, i = 1 is a valid split. - Split nums at index 2. Then, the first part is [2,3,1], and its sum is 6. The second part is [0], and its sum is 0. Since 6 >= 0, i = 2 is a valid split.

Constraints:

2 <= nums.length <= 105 -105 <= nums[i] <= 105

==
Problem Number: 1458 URL: https://leetcode.com/problems/maximum-white-tiles-covered-by-a-carpet Title: 2271. Maximum White Tiles Covered by a Carpet Problem Description: You are given a 2D integer array tiles where tiles[i] = [li, ri] represents that every tile j in the range li <= j <= ri is colored white. You are also given an integer carpetLen, the length of a single carpet that can be placed anywhere. Return the maximum number of white tiles that can be covered by the carpet. Example 1:

Input: tiles = [[1,5],[10,11],[12,18],[20,25],[30,32]], carpetLen = 10 Output: 9 Explanation: Place the carpet starting on tile 10. It covers 9 white tiles, so we return 9. Note that there may be other places where the carpet covers 9 white tiles. It can be shown that the carpet cannot cover more than 9 white tiles.

Example 2:

Input: tiles = [[10,11],[1,1]], carpetLen = 2 Output: 2 Explanation: Place the carpet starting on tile 10. It covers 2 white tiles, so we return 2.

Constraints:

1 <= tiles.length <= 5 * 104 tiles[i].length == 2 1 <= li <= ri <= 109 1 <= carpetLen <= 109 The tiles are non-overlapping.

==
Problem Number: 1459 URL: https://leetcode.com/problems/maximum-consecutive-floors-without-special-floors Title: 2274. Maximum Consecutive Floors Without Special Floors Problem Description: Alice manages a company and has rented some floors of a building as office space. Alice has decided some of these floors should be special floors, used for relaxation only. You are given two integers bottom and top, which denote that Alice has rented all the floors from bottom to top (inclusive). You are also given the integer array special, where special[i] denotes a special floor that Alice has designated for relaxation. Return the maximum number of consecutive floors without a special floor. Example 1: Input: bottom = 2, top = 9, special = [4,6] Output: 3 Explanation: The following are the ranges (inclusive) of consecutive floors without a special floor: - (2, 3) with a total amount of 2 floors. - (5, 5) with a total amount of 1 floor. - (7, 9) with a total amount of 3 floors. Therefore, we return the maximum number which is 3 floors.

Example 2: Input: bottom = 6, top = 8, special = [7,6,8] Output: 0 Explanation: Every floor rented is a special floor, so we return 0.

Constraints:

1 <= special.length <= 105 1 <= bottom <= special[i] <= top <= 109 All the values of special are unique.

==
Problem Number: 1460 URL: https://leetcode.com/problems/largest-combination-with-bitwise-and-greater-than-zero Title: 2275. Largest Combination With Bitwise AND Greater Than Zero Problem Description: The bitwise AND of an array nums is the bitwise AND of all integers in nums.

For example, for nums = [1, 5, 3], the bitwise AND is equal to 1 & 5 & 3 = 1. Also, for nums = [7], the bitwise AND is 7.

You are given an array of positive integers candidates. Evaluate the bitwise AND of every combination of numbers of candidates. Each number in candidates may only be used once in each combination. Return the size of the largest combination of candidates with a bitwise AND greater than 0. Example 1: Input: candidates = [16,17,71,62,12,24,14] Output: 4 Explanation: The combination [16,17,62,24] has a bitwise AND of 16 & 17 & 62 & 24 = 16 > 0. The size of the combination is 4. It can be shown that no combination with a size greater than 4 has a bitwise AND greater than 0. Note that more than one combination may have the largest size. For example, the combination [62,12,24,14] has a bitwise AND of 62 & 12 & 24 & 14 = 8 > 0.

Example 2: Input: candidates = [8,8] Output: 2 Explanation: The largest combination [8,8] has a bitwise AND of 8 & 8 = 8 > 0. The size of the combination is 2, so we return 2.

Constraints:

1 <= candidates.length <= 105 1 <= candidates[i] <= 107

==
Problem Number: 1461 URL: https://leetcode.com/problems/maximum-bags-with-full-capacity-of-rocks Title: 2279. Maximum Bags With Full Capacity of Rocks Problem Description: You have n bags numbered from 0 to n - 1. You are given two 0-indexed integer arrays capacity and rocks. The ith bag can hold a maximum of capacity[i] rocks and currently contains rocks[i] rocks. You are also given an integer additionalRocks, the number of additional rocks you can place in any of the bags. Return the maximum number of bags that could have full capacity after placing the additional rocks in some bags. Example 1: Input: capacity = [2,3,4,5], rocks = [1,2,4,4], additionalRocks = 2 Output: 3 Explanation: Place 1 rock in bag 0 and 1 rock in bag 1. The number of rocks in each bag are now [2,3,4,4]. Bags 0, 1, and 2 have full capacity. There are 3 bags at full capacity, so we return 3. It can be shown that it is not possible to have more than 3 bags at full capacity. Note that there may be other ways of placing the rocks that result in an answer of 3.

Example 2: Input: capacity = [10,2,2], rocks = [2,2,0], additionalRocks = 100

Output: 3 Explanation: Place 8 rocks in bag 0 and 2 rocks in bag 2. The number of rocks in each bag are now [10,2,2]. Bags 0, 1, and 2 have full capacity. There are 3 bags at full capacity, so we return 3. It can be shown that it is not possible to have more than 3 bags at full capacity. Note that we did not use all of the additional rocks.

Constraints:

n == capacity.length == rocks.length 1 <= n <= 5 * 104 1 <= capacity[i] <= 109 0 <= rocks[i] <= capacity[i] 1 <= additionalRocks <= 109

==

Problem Number: 1462 URL: https://leetcode.com/problems/minimum-lines-to-represent-a-line-chart Title: 2280. Minimum Lines to Represent a Line Chart Problem Description: You are given a 2D integer array stockPrices where stockPrices[i] = [dayi, pricei] indicates the price of the stock on day dayi is pricei. A line chart is created from the array by plotting the points on an XY plane with the X-axis representing the day and the Y-axis representing the price and connecting adjacent points. One such example is shown below:

Return the minimum number of lines needed to represent the line chart. Example 1:

Input: stockPrices = [[1,7],[2,6],[3,5],[4,4],[5,4],[6,3],[7,2],[8,1]] Output: 3 Explanation: The diagram above represents the input, with the X-axis representing the day and Y-axis representing the price. The following 3 lines can be drawn to represent the line chart: - Line 1 (in red) from (1,7) to (4,4) passing through (1,7), (2,6), (3,5), and (4,4). - Line 2 (in blue) from (4,4) to (5,4). - Line 3 (in green) from (5,4) to (8,1) passing through (5,4), (6,3), (7,2), and (8,1). It can be shown that it is not possible to represent the line chart using less than 3 lines.

Example 2:

Input: stockPrices = [[3,4],[1,2],[7,8],[2,3]] Output: 1 Explanation: As shown in the diagram above, the line chart can be represented with a single line.

Constraints:

1 <= stockPrices.length <= 105 stockPrices[i].length == 2 1 <= dayi, pricei <= 109 All dayi are distinct.

==

Problem Number: 1463 URL: https://leetcode.com/problems/sender-with-largest-word-count Title: 2284. Sender With Largest Word Count Problem Description: You have a chat log of n messages. You are given two string arrays messages and senders where messages[i] is a message sent by senders[i]. A message is list of words that are separated by a single space with no leading or trailing spaces. The word count of a sender is the total number of words sent by the sender. Note that a sender may send more than one message. Return the sender with the largest word count. If there is more than one sender with the largest word count, return the one with the lexicographically largest name. Note:

Uppercase letters come before lowercase letters in lexicographical order. "Alice" and "alice" are distinct.

Example 1: Input: messages = ["Hello userTwooo","Hi userThree","Wonderful day Alice","Nice day userThree"], senders = ["Alice","userTwo","userThree","Alice"] Output: "Alice" Explanation: Alice sends a total of 2 + 3 = 5 words. userTwo sends a total of 2 words. userThree sends a total of 3 words. Since Alice has the largest word count, we return "Alice".

Example 2: Input: messages = ["How is leetcode for everyone","Leetcode is useful for practice"], senders = ["Bob","Charlie"] Output: "Charlie" Explanation: Bob sends a total of 5 words. Charlie sends a total of 5 words. Since there is a tie for the largest word count, we return the sender with the lexicographically larger name, Charlie. Constraints:

n == messages.length == senders.length 1 <= n <= 104 1 <= messages[i].length <= 100 1 <= senders[i].length <= 10 messages[i] consists of uppercase and lowercase English letters and ' '. All the words in messages[i] are separated by a single space. messages[i] does not have leading or trailing spaces. senders[i] consists of uppercase and lowercase English letters only.

==
Problem Number: 1464 URL: https://leetcode.com/problems/maximum-total-importance-of-roads Title: 2285. Maximum Total Importance of Roads Problem Description: You are given an integer n denoting the number of cities in a country. The cities are numbered from 0 to n - 1. You are also given a 2D integer array roads where roads[i] = [ai, bi] denotes that there exists a bidirectional road connecting cities ai and bi. You need to assign each city with an integer value from 1 to n, where each value can only be used once. The importance of a road is then defined as the sum of the values of the two cities it connects. Return the maximum total importance of all roads possible after assigning the values optimally. Example 1:

Input: n = 5, roads = [[0,1],[1,2],[2,3],[0,2],[1,3],[2,4]] Output: 43 Explanation: The figure above shows the country and the assigned values of [2,4,5,3,1]. - The road (0,1) has an importance of 2 + 4 = 6. - The road (1,2) has an importance of 4 + 5 = 9. - The road (2,3) has an importance of 5 + 3 = 8. - The road (0,2) has an importance of 2 + 5 = 7. - The road (1,3) has an importance of 4 + 3 = 7. - The road (2,4) has an importance of 5 + 1 = 6. The total importance of all roads is 6 + 9 + 8 + 7 + 7 + 6 = 43. It can be shown that we cannot obtain a greater total importance than 43.

Example 2:

Input: n = 5, roads = [[0,3],[2,4],[1,3]] Output: 20 Explanation: The figure above shows the country and the assigned values of [4,3,2,5,1]. - The road (0,3) has an importance of 4 + 5 = 9. - The road (2,4) has an importance of 2 + 1

= 3. - The road (1,3) has an importance of 3 + 5 = 8. The total importance of all roads is 9 + 3 + 8 = 20. It can be shown that we cannot obtain a greater total importance than 20.

Constraints:

2 <= n <= 5 * 104 1 <= roads.length <= 5 * 104 roads[i].length == 2 0 <= ai, bi <= n - 1 ai != bi There are no duplicate roads.

==

Problem Number: 1465 URL: https://leetcode.com/problems/apply-discount-to-prices Title: 2288. Apply Discount to Prices Problem Description: A sentence is a string of single-space separated words where each word can contain digits, lowercase letters, and the dollar sign '$'. A word represents a price if it is a sequence of digits preceded by a dollar sign.

For example, "$100", "$23", and "$6" represent prices while "100", "$", and "$1e5" do not.

You are given a string sentence representing a sentence and an integer discount. For each word representing a price, apply a discount of discount% on the price and update the word in the sentence. All updated prices should be represented with exactly two decimal places. Return a string representing the modified sentence. Note that all prices will contain at most 10 digits. Example 1: Input: sentence = "there are $1 $2 and 5$ candies in the shop", discount = 50 Output: "there are $0.50 $1.00 and 5$ candies in the shop" Explanation: The words which represent prices are "$1" and "$2". - A 50% discount on "$1" yields "$0.50", so "$1" is replaced by "$0.50". - A 50% discount on "$2" yields "$1". Since we need to have exactly 2 decimal places after a price, we replace "$2" with "$1.00".

Example 2: Input: sentence = "1 2 $3 4 $5 $6 7 8$ $9 10", discount = 100 Output: "1 2 $0.00 4 $0.00 $0.00 7 8$ $0.00 10" Explanation: Applying a 100% discount on any price will result in 0. The words representing prices are "$3", "$5", "$6", and "$9". Each of them is replaced by "$0.00".

Constraints:

1 <= sentence.length <= 105 sentence consists of lowercase English letters, digits, ' ', and '$'. sentence does not have leading or trailing spaces. All words in sentence are separated by a single space. All prices will be positive numbers without leading zeros. All prices will have at most 10 digits. 0 <= discount <= 100

==

Problem Number: 1466 URL: https://leetcode.com/problems/steps-to-make-array-non-decreasing Title: 2289. Steps to Make Array Non-decreasing Problem Description: You are given a 0-indexed integer array nums. In one step, remove all elements nums[i] where nums[i - 1] > nums[i] for all 0 < i < nums.length. Return the number of steps performed until nums becomes a non-decreasing array. Example 1: Input: nums = [5,3,4,4,7,3,6,11,8,5,11] Output: 3 Explanation: The following are the steps performed: - Step 1: [5,3,4,4,7,3,6,11,8,5,11] becomes [5,4,4,7,6,11,11] - Step 2: [5,4,4,7,6,11,11] becomes [5,4,7,11,11] - Step 3: [5,4,7,11,11] becomes [5,7,11,11] [5,7,11,11] is a non-decreasing array. Therefore, we return 3.

Example 2: Input: nums = [4,5,7,7,13] Output: 0 Explanation: nums is already a non-decreasing array. Therefore, we return 0.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 1467 URL: https://leetcode.com/problems/partition-array-such-that-maximum-difference-is-k Title: 2294. Partition Array Such That Maximum Difference Is K Problem Description: You are given an integer array nums and an integer k. You may partition nums into one or more subsequences such that each element in nums appears in exactly one of the subsequences. Return the minimum number of subsequences needed such that the difference between the maximum and minimum values in each subsequence is at most k. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: nums = [3,6,1,2,5], k = 2 Output: 2 Explanation: We can partition nums into the two subsequences [3,1,2] and [6,5]. The difference between the maximum and minimum value in the first subsequence is 3 - 1 = 2. The difference between the maximum and minimum value in the second subsequence is 6 - 5 = 1. Since two subsequences were created, we return 2. It can be shown that 2 is the minimum number of subsequences needed.

Example 2: Input: nums = [1,2,3], k = 1 Output: 2 Explanation: We can partition nums into the two subsequences [1,2] and [3]. The difference between the maximum and minimum value in the first subsequence is 2 - 1 = 1. The difference between the maximum and minimum value in the second subsequence is 3 - 3 = 0. Since two subsequences were created, we return 2. Note that another optimal solution is to partition nums into the two subsequences [1] and [2,3].

Example 3: Input: nums = [2,2,4,5], k = 0 Output: 3 Explanation: We can partition nums into the three subsequences [2,2], [4], and [5]. The difference between the maximum and minimum value in the first subsequences is 2 - 2 = 0. The difference between the maximum and minimum value in the second subsequences is 4 - 4 = 0. The difference between the maximum and minimum value in the third subsequences is 5 - 5 = 0. Since three subsequences were created, we return 3. It can be shown that 3 is the minimum number of subsequences needed.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 105 0 <= k <= 105

==
Problem Number: 1468 URL: https://leetcode.com/problems/replace-elements-in-an-array Title: 2295. Replace Elements in an Array Problem Description: You are given a 0-indexed array nums that consists of n distinct positive integers. Apply m operations to this array, where in the ith operation you replace the number operations[i][0] with operations[i][1]. It is guaranteed that in the ith operation:

operations[i][0] exists in nums. operations[i][1] does not exist in nums.

Return the array obtained after applying all the operations. Example 1: Input: nums = [1,2,4,6], operations = [[1,3],[4,7],[6,1]] Output: [3,2,7,1] Explanation: We perform the following operations on nums: - Replace the number 1 with 3. nums becomes [3,2,4,6]. - Replace the number 4 with 7. nums becomes [3,2,7,6]. - Replace the number 6 with 1. nums becomes [3,2,7,1]. We return the final array [3,2,7,1].

Example 2: Input: nums = [1,2], operations = [[1,3],[2,1],[3,2]] Output: [2,1] Explanation: We perform the following operations to nums: - Replace the number 1 with 3. nums becomes [3,2]. - Replace the number 2 with 1. nums becomes [3,1]. - Replace the number 3 with 2. nums becomes [2,1]. We return the array [2,1].

Constraints:

n == nums.length m == operations.length 1 <= n, m <= 105 All the values of nums are distinct. operations[i].length == 2 1 <= nums[i], operations[i][0], operations[i][1] <= 106 operations[i][0] will exist in nums when applying the ith operation. operations[i][1] will not exist in nums when applying the ith operation.

==
Problem Number: 1469 URL: https://leetcode.com/problems/successful-pairs-of-spells-and-potions Title: 2300. Successful Pairs of Spells and Potions Problem Description: You are given two positive integer arrays spells and potions, of length n and m respectively, where spells[i] represents the strength of the ith spell and potions[j] represents the strength of the jth potion. You are also given an integer success. A spell and potion pair is considered successful if the product of their strengths is at least success. Return an integer array pairs of length n where pairs[i] is the number of potions that will form a successful pair with the ith spell. Example 1: Input: spells = [5,1,3], potions = [1,2,3,4,5], success = 7 Output: [4,0,3] Explanation: - 0th spell: 5 * [1,2,3,4,5] = [5,10,15,20,25]. 4 pairs are successful. - 1st spell: 1 * [1,2,3,4,5] = [1,2,3,4,5]. 0 pairs are successful. - 2nd spell: 3 * [1,2,3,4,5] = [3,6,9,12,15]. 3 pairs are successful. Thus, [4,0,3] is returned.

Example 2: Input: spells = [3,1,2], potions = [8,5,8], success = 16 Output: [2,0,2] Explanation: - 0th spell: 3 * [8,5,8] = [24,15,24]. 2 pairs are successful. - 1st spell: 1 * [8,5,8] = [8,5,8]. 0 pairs are successful. - 2nd spell: 2 * [8,5,8] =

[16,10,16]. 2 pairs are successful. Thus, [2,0,2] is returned.

Constraints:

n == spells.length m == potions.length 1 <= n, m <= 105 1 <= spells[i], potions[i] <= 105 1 <= success <= 1010

==
Problem Number: 1470 URL: https://leetcode.com/problems/minimum-path-cost-in-a-grid Title: 2304. Minimum Path Cost in a Grid Problem Description: You are given a 0-indexed m x n integer matrix grid consisting of distinct integers from 0 to m * n - 1. You can move in this matrix from a cell to any other cell in the next row. That is, if you are in cell (x, y) such that x < m - 1, you can move to any of the cells (x + 1, 0), (x + 1, 1), ..., (x + 1, n - 1). Note that it is not possible to move from cells in the last row. Each possible move has a cost given by a 0-indexed 2D array moveCost of size (m * n) x n, where moveCost[i][j] is the cost of moving from a cell with value i to a cell in column j of the next row. The cost of moving from cells in the last row of grid can be ignored. The cost of a path in grid is the sum of all values of cells visited plus the sum of costs of all the moves made. Return the minimum cost of a path that starts from any cell in the first row and ends at any cell in the last row. Example 1:

Input: grid = [[5,3],[4,0],[2,1]], moveCost = [[9,8],[1,5],[10,12],[18,6],[2,4],[14,3]] Output: 17 Explanation: The path with the minimum possible cost is the path 5 -> 0 -> 1. - The sum of the values of cells visited is 5 + 0 + 1 = 6. - The cost of moving from 5 to 0 is 3. - The cost of moving from 0 to 1 is 8. So the total cost of the path is 6 + 3 + 8 = 17.

Example 2: Input: grid = [[5,1,2],[4,0,3]], moveCost = [[12,10,15],[20,23,8],[21,7,1],[8,1,13],[9,10,25],[5,3,2]] Output: 6 Explanation: The path with the minimum possible cost is the path 2 -> 3. - The sum of the values of cells visited is 2 + 3 = 5. - The cost of moving from 2 to 3 is 1. So the total cost of this path is 5 + 1 = 6.

Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 50 grid consists of distinct integers from 0 to m * n - 1. moveCost.length == m * n moveCost[i].length == n 1 <= moveCost[i][j] <= 100

==
Problem Number: 1471 URL: https://leetcode.com/problems/fair-distribution-of-cookies Title: 2305. Fair Distribution of Cookies Problem Description: You are given an integer array cookies, where cookies[i] denotes the number of cookies in the ith bag. You are also given an integer k that denotes the number of children to distribute all the bags of cookies to. All the cookies in the same bag must go to the same child and cannot be split up. The unfairness of a distribution is defined as the maximum total cookies obtained by a single child in the distribution. Return the minimum unfairness of all distributions. Example 1: Input: cookies = [8,15,10,20,8], k = 2 Output: 31 Explanation:

One optimal distribution is [8,15,8] and [10,20] - The 1st child receives [8,15,8] which has a total of 8 + 15 + 8 = 31 cookies. - The 2nd child receives [10,20] which has a total of 10 + 20 = 30 cookies. The unfairness of the distribution is max(31,30) = 31. It can be shown that there is no distribution with an unfairness less than 31.

Example 2: Input: cookies = [6,1,3,2,2,4,1,2], k = 3 Output: 7 Explanation: One optimal distribution is [6,1], [3,2,2], and [4,1,2] - The 1st child receives [6,1] which has a total of 6 + 1 = 7 cookies. - The 2nd child receives [3,2,2] which has a total of 3 + 2 + 2 = 7 cookies. - The 3rd child receives [4,1,2] which has a total of 4 + 1 + 2 = 7 cookies. The unfairness of the distribution is max(7,7,7) = 7. It can be shown that there is no distribution with an unfairness less than 7.

Constraints:

2 <= cookies.length <= 8 1 <= cookies[i] <= 105 2 <= k <= cookies.length

==
Problem Number: 1472 URL: https://leetcode.com/problems/sum-of-numbers-with-units-digit-k Title: 2310. Sum of Numbers With Units Digit K Problem Description: Given two integers num and k, consider a set of positive integers with the following properties:

The units digit of each integer is k. The sum of the integers is num.

Return the minimum possible size of such a set, or -1 if no such set exists. Note:

The set can contain multiple instances of the same integer, and the sum of an empty set is considered 0. The units digit of a number is the rightmost digit of the number.

Example 1: Input: num = 58, k = 9 Output: 2 Explanation: One valid set is [9,49], as the sum is 58 and each integer has a units digit of 9. Another valid set is [19,39]. It can be shown that 2 is the minimum possible size of a valid set.

Example 2: Input: num = 37, k = 2 Output: -1 Explanation: It is not possible to obtain a sum of 37 using only integers that have a units digit of 2.

Example 3: Input: num = 0, k = 7 Output: 0 Explanation: The sum of an empty set is considered 0.

Constraints:

0 <= num <= 3000 0 <= k <= 9

==
Problem Number: 1473 URL: https://leetcode.com/problems/longest-binary-subsequence-less-than-or-equal-to-k Title: 2311. Longest Binary Subsequence Less Than or Equal to K Problem Description: You are given a binary string s and a positive integer k. Return the length of the longest subsequence of s that makes up a binary number less than or equal to k. Note:

The subsequence can contain leading zeroes. The empty string is considered to be equal to 0. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

Example 1: Input: s = "1001010", k = 5 Output: 5 Explanation: The longest subsequence of s that makes up a binary number less than or equal to 5 is "00010", as this number is equal to 2 in decimal. Note that "00100" and "00101" are also possible, which are equal to 4 and 5 in decimal, respectively. The length of this subsequence is 5, so 5 is returned.

Example 2: Input: s = "00101001", k = 1 Output: 6 Explanation: "000001" is the longest subsequence of s that makes up a binary number less than or equal to 1, as this number is equal to 1 in decimal. The length of this subsequence is 6, so 6 is returned.

Constraints:

1 <= s.length <= 1000 s[i] is either '0' or '1'. 1 <= k <= 109

==
Problem Number: 1474 URL: https://leetcode.com/problems/count-unreachable-pairs-of-nodes-in-an-undirected-graph Title: 2316. Count Unreachable Pairs of Nodes in an Undirected Graph Problem Description: You are given an integer n. There is an undirected graph with n nodes, numbered from 0 to n - 1. You are given a 2D integer array edges where edges[i] = [ai, bi] denotes that there exists an undirected edge connecting nodes ai and bi. Return the number of pairs of different nodes that are unreachable from each other. Example 1:

Input: n = 3, edges = [[0,1],[0,2],[1,2]] Output: 0 Explanation: There are no pairs of nodes that are unreachable from each other. Therefore, we return 0.

Example 2:

Input: n = 7, edges = [[0,2],[0,5],[2,4],[1,6],[5,4]] Output: 14 Explanation: There are 14 pairs of nodes that are unreachable from each other: [[0,1],[0,3],[0,6],[1,2],[1,3],[1,4],[1,5],[2,3],[2,6],[3,4],[3,5],[3,6],[4,6],[5,6]]. Therefore, we return 14.

Constraints:

1 <= n <= 105 0 <= edges.length <= 2 * 105 edges[i].length == 2 0 <= ai, bi < n ai != bi There are no repeated edges.

==
Problem Number: 1475 URL: https://leetcode.com/problems/maximum-xor-after-operations Title: 2317. Maximum XOR After Operations Problem Description: You are given a 0-indexed integer array nums. In one operation, select any non-negative integer x and an index i, then update nums[i] to be equal to nums[i] AND (nums[i] XOR x). Note that AND is the bitwise AND

operation and XOR is the bitwise XOR operation. Return the maximum possible bitwise XOR of all elements of nums after applying the operation any number of times. Example 1: Input: nums = [3,2,4,6] Output: 7 Explanation: Apply the operation with x = 4 and i = 3, num[3] = 6 AND (6 XOR 4) = 6 AND 2 = 2. Now, nums = [3, 2, 4, 2] and the bitwise XOR of all the elements = 3 XOR 2 XOR 4 XOR 2 = 7. It can be shown that 7 is the maximum possible bitwise XOR. Note that other operations may be used to achieve a bitwise XOR of 7. Example 2: Input: nums = [1,2,3,9,2] Output: 11 Explanation: Apply the operation zero times. The bitwise XOR of all the elements = 1 XOR 2 XOR 3 XOR 9 XOR 2 = 11. It can be shown that 11 is the maximum possible bitwise XOR. Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 108

==
Problem Number: 1476 URL: https://leetcode.com/problems/count-number-of-ways-to-place-houses Title: 2320. Count Number of Ways to Place Houses Problem Description: There is a street with n * 2 plots, where there are n plots on each side of the street. The plots on each side are numbered from 1 to n. On each plot, a house can be placed. Return the number of ways houses can be placed such that no two houses are adjacent to each other on the same side of the street. Since the answer may be very large, return it modulo 109 + 7. Note that if a house is placed on the ith plot on one side of the street, a house can also be placed on the ith plot on the other side of the street. Example 1: Input: n = 1 Output: 4 Explanation: Possible arrangements: 1. All plots are empty. 2. A house is placed on one side of the street. 3. A house is placed on the other side of the street. 4. Two houses are placed, one on each side of the street.

Example 2:

Input: n = 2 Output: 9 Explanation: The 9 possible arrangements are shown in the diagram above.

Constraints:

1 <= n <= 104

==
Problem Number: 1477 URL: https://leetcode.com/problems/spiral-matrix-iv Title: 2326. Spiral Matrix IV Problem Description: You are given two integers m and n, which represent the dimensions of a matrix. You are also given the head of a linked list of integers. Generate an m x n matrix that contains the integers in the linked list presented in spiral order (clockwise), starting from the top-left of the matrix. If there are remaining empty spaces, fill them with -1. Return the generated matrix. Example 1:

Input: m = 3, n = 5, head = [3,0,2,6,8,1,7,9,4,2,5,5,0] Output: [[3,0,2,6,8],[5,0,-1,-1,1],[5,2,4,9,7]] Explanation: The diagram above shows how the values are printed in the matrix. Note that the remaining spaces in the matrix are filled with -1.

Example 2:

Input: m = 1, n = 4, head = [0,1,2] Output: [[0,1,2,-1]] Explanation: The diagram above shows how the values are printed from left to right in the matrix. The last space in the matrix is set to -1. Constraints:

1 <= m, n <= 105 1 <= m * n <= 105 The number of nodes in the list is in the range [1, m * n]. 0 <= Node.val <= 1000

==
Problem Number: 1478 URL: https://leetcode.com/problems/number-of-people-aware-of-a-secret Title: 2327. Number of People Aware of a Secret Problem Description: On day 1, one person discovers a secret. You are given an integer delay, which means that each person will share the secret with a new person every day, starting from delay days after discovering the secret. You are also given an integer forget, which means that each person will forget the secret forget days after discovering it. A person cannot share the secret on the same day they forgot it, or on any day afterwards. Given an integer n, return the number of people who know the secret at the end of day n. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: n = 6, delay = 2, forget = 4 Output: 5 Explanation: Day 1: Suppose the first person is named A. (1 person) Day 2: A is the only person who knows the secret. (1 person) Day 3: A shares the secret with a new person, B. (2 people) Day 4: A shares the secret with a new person, C. (3 people) Day 5: A forgets the secret, and B shares the secret with a new person, D. (3 people) Day 6: B shares the secret with E, and C shares the secret with F. (5 people)

Example 2: Input: n = 4, delay = 1, forget = 3 Output: 6 Explanation: Day 1: The first person is named A. (1 person) Day 2: A shares the secret with B. (2 people) Day 3: A and B share the secret with 2 new people, C and D. (4 people) Day 4: A forgets the secret. B, C, and D share the secret with 3 new people. (6 people)

Constraints:

2 <= n <= 1000 1 <= delay < forget <= n

==
Problem Number: 1479 URL: https://leetcode.com/problems/the-latest-time-to-catch-a-bus Title: 2332. The Latest Time to Catch a Bus Problem Description: You are given a 0-indexed integer array buses of length n, where buses[i] represents the departure time of the ith bus. You are also given a 0-indexed integer array passengers of length m, where passengers[j] represents the arrival time of the jth passenger. All bus departure times are unique. All passenger arrival times are unique. You are given an integer capacity, which represents the maximum number of passengers that can get on each bus. When a passenger arrives, they will wait in line for the next available bus. You can

get on a bus that departs at x minutes if you arrive at y minutes where y <= x, and the bus is not full. Passengers with the earliest arrival times get on the bus first. More formally when a bus arrives, either:

If capacity or fewer passengers are waiting for a bus, they will all get on the bus, or The capacity passengers with the earliest arrival times will get on the bus.

Return the latest time you may arrive at the bus station to catch a bus. You cannot arrive at the same time as another passenger. Note: The arrays buses and passengers are not necessarily sorted.   Example 1: Input: buses = [10,20], passengers = [2,17,18,19], capacity = 2 Output: 16 Explanation: Suppose you arrive at time 16. At time 10, the first bus departs with the 0th passenger. At time 20, the second bus departs with you and the 1st passenger. Note that you may not arrive at the same time as another passenger, which is why you must arrive before the 1st passenger to catch the bus. Example 2: Input: buses = [20,30,10], passengers = [19,13,26,4,25,11,21], capacity = 2 Output: 20 Explanation: Suppose you arrive at time 20. At time 10, the first bus departs with the 3rd passenger. At time 20, the second bus departs with the 5th and 1st passengers. At time 30, the third bus departs with the 0th passenger and you. Notice if you had arrived any later, then the 6th passenger would have taken your seat on the third bus.   Constraints:

n == buses.length m == passengers.length 1 <= n, m, capacity <= 105 2 <= buses[i], passengers[i] <= 109 Each element in buses is unique. Each element in passengers is unique.

==
Problem Number: 1480 URL: https://leetcode.com/problems/minimum-sum-of-squared-difference Title: 2333. Minimum Sum of Squared Difference Problem Description: You are given two positive 0-indexed integer arrays nums1 and nums2, both of length n. The sum of squared difference of arrays nums1 and nums2 is defined as the sum of (nums1[i] - nums2[i])2 for each 0 <= i < n. You are also given two positive integers k1 and k2. You can modify any of the elements of nums1 by +1 or -1 at most k1 times. Similarly, you can modify any of the elements of nums2 by +1 or -1 at most k2 times. Return the minimum sum of squared difference after modifying array nums1 at most k1 times and modifying array nums2 at most k2 times. Note: You are allowed to modify the array elements to become negative integers.   Example 1: Input: nums1 = [1,2,3,4], nums2 = [2,10,20,19], k1 = 0, k2 = 0 Output: 579 Explanation: The elements in nums1 and nums2 cannot be modified because k1 = 0 and k2 = 0. The sum of square difference will be: (1 - 2)2 + (2 - 10)2 + (3 - 20)2 + (4 - 19)2 = 579.

Example 2: Input: nums1 = [1,4,10,12], nums2 = [5,8,6,9], k1 = 1, k2 = 1 Output: 43 Explanation: One way to obtain the minimum sum of square difference is: - Increase nums1[0] once. - Increase nums2[2] once. The minimum of the sum of square difference will be: (2 - 5)2 + (4 - 8)2 + (10 - 7)2 + (12 - 9)2 = 43. Note that, there are other ways to obtain the minimum of the sum of square difference, but there is no way to obtain a sum smaller than 43. Constraints:

n == nums1.length == nums2.length 1 <= n <= 105 0 <= nums1[i], nums2[i] <= 105 0 <= k1, k2 <= 109

==
Problem Number: 1481 URL: https://leetcode.com/problems/smallest-number-in-infinite-set Title: 2336. Smallest Number in Infinite Set Problem Description: You have a set which contains all positive integers [1, 2, 3, 4, 5, ...]. Implement the SmallestInfiniteSet class:

SmallestInfiniteSet() Initializes the SmallestInfiniteSet object to contain all positive integers. int popSmallest() Removes and returns the smallest integer contained in the infinite set. void addBack(int num) Adds a positive integer num back into the infinite set, if it is not already in the infinite set.

Example 1: Input ["SmallestInfiniteSet", "addBack", "popSmallest", "popSmallest", "popSmallest", "addBack", "popSmallest", "popSmallest", "popSmallest"] [[], [2], [], [], [], [1], [], [], []] Output [null, null, 1, 2, 3, null, 1, 4, 5]

Explanation SmallestInfiniteSet smallestInfiniteSet = new SmallestInfiniteSet(); smallestInfiniteSet.addBack(2); // 2 is already in the set, so no change is made. smallestInfiniteSet.popSmallest(); // return 1, since 1 is the smallest number, and remove it from the set. smallestInfiniteSet.popSmallest(); // return 2, and remove it from the set. smallestInfiniteSet.popSmallest(); // return 3, and remove it from the set. smallestInfiniteSet.addBack(1); // 1 is added back to the set. smallestInfiniteSet.popSmallest(); // return 1, since 1 was added back to the set and // is the smallest number, and remove it from the set. smallestInfiniteSet.popSmallest(); // return 4, and remove it from the set. smallestInfiniteSet.popSmallest(); // return 5, and remove it from the set.

Constraints:

1 <= num <= 1000 At most 1000 calls will be made in total to popSmallest and addBack.

==
Problem Number: 1482 URL: https://leetcode.com/problems/move-pieces-to-obtain-a-string Title: 2337. Move Pieces to Obtain a String Problem Description: You are given two strings start and target, both of length n. Each string consists only of the characters 'L', 'R', and '_' where:

The characters 'L' and 'R' represent pieces, where a piece 'L' can move to the left only if there is a blank space directly to its left, and a piece 'R' can move to the right only if there is a blank space directly to its right. The character '_' represents a blank space that can be occupied by any of the 'L' or 'R' pieces.

Return true if it is possible to obtain the string target by moving the pieces of the string start any number of times. Otherwise, return false. Example 1: Input: start = "_L__R__R_", target = "L______RR" Output: true

Explanation: We can obtain the string target from start by doing the following moves: - Move the first piece one step to the left, start becomes equal to "L___R__R_". - Move the last piece one step to the right, start becomes equal to "L___R___R". - Move the second piece three steps to the right, start becomes equal to "L______RR". Since it is possible to get the string target from start, we return true.

Example 2: Input: start = "R_L_", target = "__LR" Output: false Explanation: The 'R' piece in the string start can move one step to the right to obtain "_RL_". After that, no pieces can move anymore, so it is impossible to obtain the string target from start.

Example 3: Input: start = "_R", target = "R_" Output: false Explanation: The piece in the string start can move only to the right, so it is impossible to obtain the string target from start. Constraints:

n == start.length == target.length 1 <= n <= 105 start and target consist of the characters 'L', 'R', and '_'.

==

Problem Number: 1483 URL: https://leetcode.com/problems/max-sum-of-a-pair-with-equal-sum-of-digits Title: 2342. Max Sum of a Pair With Equal Sum of Digits Problem Description: You are given a 0-indexed array nums consisting of positive integers. You can choose two indices i and j, such that i != j, and the sum of digits of the number nums[i] is equal to that of nums[j]. Return the maximum value of nums[i] + nums[j] that you can obtain over all possible indices i and j that satisfy the conditions. Example 1: Input: nums = [18,43,36,13,7] Output: 54 Explanation: The pairs (i, j) that satisfy the conditions are: - (0, 2), both numbers have a sum of digits equal to 9, and their sum is 18 + 36 = 54. - (1, 4), both numbers have a sum of digits equal to 7, and their sum is 43 + 7 = 50. So the maximum sum that we can obtain is 54.

Example 2: Input: nums = [10,12,19,14] Output: -1 Explanation: There are no two numbers that satisfy the conditions, so we return -1.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 1484 URL: https://leetcode.com/problems/query-kth-smallest-trimmed-number Title: 2343. Query Kth Smallest Trimmed Number Problem Description: You are given a 0-indexed array of strings nums, where each string is of equal length and consists of only digits. You are also given a 0-indexed 2D integer array queries where queries[i] = [ki, trimi]. For each queries[i], you need to:

Trim each number in nums to its rightmost trimi digits. Determine the index of the kith smallest trimmed number in nums. If two trimmed numbers are equal, the number with the lower index is considered to be smaller. Reset each number in nums to its original length.

Return an array answer of the same length as queries, where answer[i] is the answer to the ith query. Note:

To trim to the rightmost x digits means to keep removing the leftmost digit, until only x digits remain. Strings in nums may contain leading zeros.

Example 1: Input: nums = ["102","473","251","814"], queries = [[1,1],[2,3],[4,2],[1,2]] Output: [2,2,1,0] Explanation: 1. After trimming to the last digit, nums = ["2","3","1","4"]. The smallest number is 1 at index 2. 2. Trimmed to the last 3 digits, nums is unchanged. The 2nd smallest number is 251 at index 2. 3. Trimmed to the last 2 digits, nums = ["02","73","51","14"]. The 4th smallest number is 73. 4. Trimmed to the last 2 digits, the smallest number is 2 at index 0. Note that the trimmed number "02" is evaluated as 2.

Example 2: Input: nums = ["24","37","96","04"], queries = [[2,1],[2,2]] Output: [3,0] Explanation: 1. Trimmed to the last digit, nums = ["4","7","6","4"]. The 2nd smallest number is 4 at index 3. There are two occurrences of 4, but the one at index 0 is considered smaller than the one at index 3. 2. Trimmed to the last 2 digits, nums is unchanged. The 2nd smallest number is 24.

Constraints:

1 <= nums.length <= 100 1 <= nums[i].length <= 100 nums[i] consists of only digits. All nums[i].length are equal. 1 <= queries.length <= 100 queries[i].length == 2 1 <= ki <= nums.length 1 <= trimi <= nums[i].length

Follow up: Could you use the Radix Sort Algorithm to solve this problem? What will be the complexity of that solution?

==
Problem Number: 1485 URL: https://leetcode.com/problems/number-of-zero-filled-subarrays Title: 2348. Number of Zero-Filled Subarrays Problem Description: Given an integer array nums, return the number of subarrays filled with 0. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,3,0,0,2,0,0,4] Output: 6 Explanation: There are 4 occurrences of [0] as a subarray. There are 2 occurrences of [0,0] as a subarray. There is no occurrence of a subarray with a size more than 2 filled with 0. Therefore, we return 6. Example 2: Input: nums = [0,0,0,2,0,0] Output: 9 Explanation: There are 5 occurrences of [0] as a subarray. There are 3 occurrences of [0,0] as a subarray. There is 1 occurrence of [0,0,0] as a subarray. There is no occurrence of a subarray with a size more than 3 filled with 0. Therefore, we return 9.

Example 3: Input: nums = [2,10,2019] Output: 0 Explanation: There is no subarray filled with 0. Therefore, we return 0.

Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109

==

Problem Number: 1486 URL: https://leetcode.com/problems/design-a-number-container-system Title: 2349. Design a Number Container System Problem Description: Design a number container system that can do the following:

Insert or Replace a number at the given index in the system. Return the smallest index for the given number in the system.

Implement the NumberContainers class:

NumberContainers() Initializes the number container system. void change(int index, int number) Fills the container at index with the number. If there is already a number at that index, replace it. int find(int number) Returns the smallest index for the given number, or -1 if there is no index that is filled by number in the system.

Example 1: Input ["NumberContainers", "find", "change", "change", "change", "change", "find", "change", "find"] [[], [10], [2, 10], [1, 10], [3, 10], [5, 10], [10], [1, 20], [10]] Output [null, -1, null, null, null, null, 1, null, 2]

Explanation NumberContainers nc = new NumberContainers(); nc.find(10); // There is no index that is filled with number 10. Therefore, we return -1. nc.change(2, 10); // Your container at index 2 will be filled with number 10. nc.change(1, 10); // Your container at index 1 will be filled with number 10. nc.change(3, 10); // Your container at index 3 will be filled with number 10. nc.change(5, 10); // Your container at index 5 will be filled with number 10. nc.find(10); // Number 10 is at the indices 1, 2, 3, and 5. Since the smallest index that is filled with 10 is 1, we return 1. nc.change(1, 20); // Your container at index 1 will be filled with number 20. Note that index 1 was filled with 10 and then replaced with 20. nc.find(10); // Number 10 is at the indices 2, 3, and 5. The smallest index that is filled with 10 is 2. Therefore, we return 2.

Constraints:

1 <= index, number <= 109 At most 105 calls will be made in total to change and find.

==

Problem Number: 1487 URL: https://leetcode.com/problems/equal-row-and-column-pairs Title: 2352. Equal Row and Column Pairs Problem Description: Given a 0-indexed n x n integer matrix grid, return the number of pairs (ri, cj) such that row ri and column cj are equal. A row and column pair is considered equal if they contain the same elements in the same order (i.e., an equal array). Example 1:

Input: grid = [[3,2,1],[1,7,6],[2,7,7]] Output: 1 Explanation: There is 1 equal row and column pair: - (Row 2, Column 1): [2,7,7]

Example 2:

Input: grid = [[3,1,2,2],[1,4,4,5],[2,4,2,2],[2,4,2,2]] Output: 3 Explanation: There are 3 equal row and column pairs: - (Row 0, Column 0): [3,1,2,2] - (Row 2, Column 2): [2,4,2,2] - (Row 3, Column 2): [2,4,2,2]

Constraints:

n == grid.length == grid[i].length 1 <= n <= 200 1 <= grid[i][j] <= 105

==

Problem Number: 1488 URL: https://leetcode.com/problems/design-a-food-rating-system Title: 2353. Design a Food Rating System Problem Description: Design a food rating system that can do the following:

Modify the rating of a food item listed in the system. Return the highest-rated food item for a type of cuisine in the system.

Implement the FoodRatings class:

FoodRatings(String[] foods, String[] cuisines, int[] ratings) Initializes the system. The food items are described by foods, cuisines and ratings, all of which have a length of n.

foods[i] is the name of the ith food, cuisines[i] is the type of cuisine of the ith food, and ratings[i] is the initial rating of the ith food.

void changeRating(String food, int newRating) Changes the rating of the food item with the name food. String highestRated(String cuisine) Returns the name of the food item that has the highest rating for the given type of cuisine. If there is a tie, return the item with the lexicographically smaller name.

Note that a string x is lexicographically smaller than string y if x comes before y in dictionary order, that is, either x is a prefix of y, or if i is the first position such that x[i] != y[i], then x[i] comes before y[i] in alphabetic order. Example 1: Input ["FoodRatings", "highestRated", "highestRated", "changeRating", "highestRated", "changeRating", "highestRated"] [[["kimchi", "miso", "sushi", "moussaka", "ramen", "bulgogi"], ["korean", "japanese", "japanese", "greek", "japanese", "korean"], [9, 12, 8, 15, 14, 7]], ["korean"], ["japanese"], ["sushi", 16], ["japanese"], ["ramen", 16], ["japanese"]] Output [null, "kimchi", "ramen", null, "sushi", null, "ramen"]

Explanation FoodRatings foodRatings = new FoodRatings(["kimchi", "miso", "sushi", "moussaka", "ramen", "bulgogi"], ["korean", "japanese", "japanese", "greek", "japanese", "korean"], [9, 12, 8, 15, 14, 7]); foodRatings.highestRated("korean"); // return "kimchi" // "kimchi" is the highest rated korean food with a rating of 9. foodRatings.highestRated("japanese"); // return "ramen" // "ramen" is the highest rated japanese food with a rating of 14. foodRatings.changeRating("sushi", 16); // "sushi" now has a rating of 16. foodRatings.highestRated("japanese"); // return "sushi" // "sushi" is the highest rated japanese food with a rating of 16. foodRatings.changeRating("ramen", 16); // "ramen" now has a rating of 16. foodRatings.highestRated("japanese");

// return ”ramen” // Both ”sushi” and ”ramen” have a rating of 16. // However, ”ramen” is lexicographically smaller than ”sushi”.

Constraints:

1 <= n <= 2 * 104 n == foods.length == cuisines.length == ratings.length 1 <= foods[i].length, cuisines[i].length <= 10 foods[i], cuisines[i] consist of lowercase English letters. 1 <= ratings[i] <= 108 All the strings in foods are distinct. food will be the name of a food item in the system across all calls to changeRating. cuisine will be a type of cuisine of at least one food item in the system across all calls to highestRated. At most 2 * 104 calls in total will be made to changeRating and highestRated.

==
Problem Number: 1489 URL: https://leetcode.com/problems/maximum-number-of-groups-entering-a-competition Title: 2358. Maximum Number of Groups Entering a Competition Problem Description: You are given a positive integer array grades which represents the grades of students in a university. You would like to enter all these students into a competition in ordered non-empty groups, such that the ordering meets the following conditions:

The sum of the grades of students in the ith group is less than the sum of the grades of students in the (i + 1)th group, for all groups (except the last). The total number of students in the ith group is less than the total number of students in the (i + 1)th group, for all groups (except the last).

Return the maximum number of groups that can be formed. Example 1: Input: grades = [10,6,12,7,3,5] Output: 3 Explanation: The following is a possible way to form 3 groups of students: - 1st group has the students with grades = [12]. Sum of grades: 12. Student count: 1 - 2nd group has the students with grades = [6,7]. Sum of grades: 6 + 7 = 13. Student count: 2 - 3rd group has the students with grades = [10,3,5]. Sum of grades: 10 + 3 + 5 = 18. Student count: 3 It can be shown that it is not possible to form more than 3 groups.

Example 2: Input: grades = [8,8] Output: 1 Explanation: We can only form 1 group, since forming 2 groups would lead to an equal number of students in both groups.

Constraints:

1 <= grades.length <= 105 1 <= grades[i] <= 105

==
Problem Number: 1490 URL: https://leetcode.com/problems/find-closest-node-to-given-two-nodes Title: 2359. Find Closest Node to Given Two Nodes Problem Description: You are given a directed graph of n nodes numbered from 0 to n - 1, where each node has at most one outgoing edge. The graph is represented with a given 0-indexed array edges of size n, indicating that there is a directed edge from node i to node edges[i]. If there is no outgoing edge from i, then edges[i] == -1. You are also given two integers node1 and node2.

Return the index of the node that can be reached from both node1 and node2, such that the maximum between the distance from node1 to that node, and from node2 to that node is minimized. If there are multiple answers, return the node with the smallest index, and if no possible answer exists, return -1. Note that edges may contain cycles.   Example 1:

Input: edges = [2,2,3,-1], node1 = 0, node2 = 1 Output: 2 Explanation: The distance from node 0 to node 2 is 1, and the distance from node 1 to node 2 is 1. The maximum of those two distances is 1. It can be proven that we cannot get a node with a smaller maximum distance than 1, so we return node 2.

Example 2:

Input: edges = [1,2,-1], node1 = 0, node2 = 2 Output: 2 Explanation: The distance from node 0 to node 2 is 2, and the distance from node 2 to itself is 0. The maximum of those two distances is 2. It can be proven that we cannot get a node with a smaller maximum distance than 2, so we return node 2.

Constraints:

n == edges.length 2 <= n <= 105 -1 <= edges[i] < n edges[i] != i 0 <= node1, node2 < n

==
Problem Number: 1491 URL: https://leetcode.com/problems/count-number-of-bad-pairs Title: 2364. Count Number of Bad Pairs Problem Description: You are given a 0-indexed integer array nums. A pair of indices (i, j) is a bad pair if i < j and j - i != nums[j] - nums[i]. Return the total number of bad pairs in nums.   Example 1: Input: nums = [4,1,3,3] Output: 5 Explanation: The pair (0, 1) is a bad pair since 1 - 0 != 1 - 4. The pair (0, 2) is a bad pair since 2 - 0 != 3 - 4, 2 != -1. The pair (0, 3) is a bad pair since 3 - 0 != 3 - 4, 3 != -1. The pair (1, 2) is a bad pair since 2 - 1 != 3 - 1, 1 != 2. The pair (2, 3) is a bad pair since 3 - 2 != 3 - 3, 1 != 0. There are a total of 5 bad pairs, so we return 5.

Example 2: Input: nums = [1,2,3,4,5] Output: 0 Explanation: There are no bad pairs.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1492 URL: https://leetcode.com/problems/task-scheduler-ii Title: 2365. Task Scheduler II Problem Description: You are given a 0-indexed array of positive integers tasks, representing tasks that need to be completed in order, where tasks[i] represents the type of the ith task. You are also given a positive integer space, which represents the minimum number of days that must pass after the completion of a task before another task of the same type can be performed. Each day, until all tasks have been completed, you must either:

Complete the next task from tasks, or Take a break.

Return the minimum number of days needed to complete all tasks. Example 1: Input: tasks = [1,2,1,2,3,1], space = 3 Output: 9 Explanation: One way to complete all tasks in 9 days is as follows: Day 1: Complete the 0th task. Day 2: Complete the 1st task. Day 3: Take a break. Day 4: Take a break. Day 5: Complete the 2nd task. Day 6: Complete the 3rd task. Day 7: Take a break. Day 8: Complete the 4th task. Day 9: Complete the 5th task. It can be shown that the tasks cannot be completed in less than 9 days.

Example 2: Input: tasks = [5,8,8,5], space = 2 Output: 6 Explanation: One way to complete all tasks in 6 days is as follows: Day 1: Complete the 0th task. Day 2: Complete the 1st task. Day 3: Take a break. Day 4: Take a break. Day 5: Complete the 2nd task. Day 6: Complete the 3rd task. It can be shown that the tasks cannot be completed in less than 6 days.

Constraints:

1 <= tasks.length <= 105 1 <= tasks[i] <= 109 1 <= space <= tasks.length

==
Problem Number: 1493 URL: https://leetcode.com/problems/reachable-nodes-with-restrictions Title: 2368. Reachable Nodes With Restrictions Problem Description: There is an undirected tree with n nodes labeled from 0 to n - 1 and n - 1 edges. You are given a 2D integer array edges of length n - 1 where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. You are also given an integer array restricted which represents restricted nodes. Return the maximum number of nodes you can reach from node 0 without visiting a restricted node. Note that node 0 will not be a restricted node. Example 1:

Input: n = 7, edges = [[0,1],[1,2],[3,1],[4,0],[0,5],[5,6]], restricted = [4,5] Output: 4 Explanation: The diagram above shows the tree. We have that [0,1,2,3] are the only nodes that can be reached from node 0 without visiting a restricted node.

Example 2:

Input: n = 7, edges = [[0,1],[0,2],[0,5],[0,4],[3,2],[6,5]], restricted = [4,2,1] Output: 3 Explanation: The diagram above shows the tree. We have that [0,5,6] are the only nodes that can be reached from node 0 without visiting a restricted node.

Constraints:

2 <= n <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi edges represents a valid tree. 1 <= restricted.length < n 1 <= restricted[i] < n All the values of restricted are unique.

==
Problem Number: 1494 URL: https://leetcode.com/problems/check-if-there-is-a-valid-partition-for-the-array Title: 2369. Check if There is a Valid Partition

For The Array Problem Description: You are given a 0-indexed integer array nums. You have to partition the array into one or more contiguous subarrays. We call a partition of the array valid if each of the obtained subarrays satisfies one of the following conditions:

The subarray consists of exactly 2, equal elements. For example, the subarray [2,2] is good. The subarray consists of exactly 3, equal elements. For example, the subarray [4,4,4] is good. The subarray consists of exactly 3 consecutive increasing elements, that is, the difference between adjacent elements is 1. For example, the subarray [3,4,5] is good, but the subarray [1,3,5] is not.

Return true if the array has at least one valid partition. Otherwise, return false. Example 1: Input: nums = [4,4,4,5,6] Output: true Explanation: The array can be partitioned into the subarrays [4,4] and [4,5,6]. This partition is valid, so we return true.

Example 2: Input: nums = [1,1,1,2] Output: false Explanation: There is no valid partition for this array.

Constraints:

2 <= nums.length <= 105 1 <= nums[i] <= 106

==

Problem Number: 1495 URL: https://leetcode.com/problems/longest-ideal-subsequence Title: 2370. Longest Ideal Subsequence Problem Description: You are given a string s consisting of lowercase letters and an integer k. We call a string t ideal if the following conditions are satisfied:

t is a subsequence of the string s. The absolute difference in the alphabet order of every two adjacent letters in t is less than or equal to k.

Return the length of the longest ideal string. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. Note that the alphabet order is not cyclic. For example, the absolute difference in the alphabet order of 'a' and 'z' is 25, not 1. Example 1: Input: s = "acfgbd", k = 2 Output: 4 Explanation: The longest ideal string is "acbd". The length of this string is 4, so 4 is returned. Note that "acfgbd" is not ideal because 'c' and 'f' have a difference of 3 in alphabet order. Example 2: Input: s = "abcd", k = 3 Output: 4 Explanation: The longest ideal string is "abcd". The length of this string is 4, so 4 is returned.

Constraints:

1 <= s.length <= 105 0 <= k <= 25 s consists of lowercase English letters.

==

Problem Number: 1496 URL: https://leetcode.com/problems/node-with-highest-edge-score Title: 2374. Node With Highest Edge Score Problem Description: You are given a directed graph with n nodes labeled from 0 to n -

1, where each node has exactly one outgoing edge. The graph is represented by a given 0-indexed integer array edges of length n, where edges[i] indicates that there is a directed edge from node i to node edges[i]. The edge score of a node i is defined as the sum of the labels of all the nodes that have an edge pointing to i. Return the node with the highest edge score. If multiple nodes have the same edge score, return the node with the smallest index. Example 1:

Input: edges = [1,0,0,0,0,7,7,5] Output: 7 Explanation: - The nodes 1, 2, 3 and 4 have an edge pointing to node 0. The edge score of node 0 is 1 + 2 + 3 + 4 = 10. - The node 0 has an edge pointing to node 1. The edge score of node 1 is 0. - The node 7 has an edge pointing to node 5. The edge score of node 5 is 7. - The nodes 5 and 6 have an edge pointing to node 7. The edge score of node 7 is 5 + 6 = 11. Node 7 has the highest edge score so return 7.

Example 2:

Input: edges = [2,0,0,2] Output: 0 Explanation: - The nodes 1 and 2 have an edge pointing to node 0. The edge score of node 0 is 1 + 2 = 3. - The nodes 0 and 3 have an edge pointing to node 2. The edge score of node 2 is 0 + 3 = 3. Nodes 0 and 2 both have an edge score of 3. Since node 0 has a smaller index, we return 0.

Constraints:

n == edges.length 2 <= n <= 105 0 <= edges[i] < n edges[i] != i

==

Problem Number: 1497 URL: https://leetcode.com/problems/construct-smallest-number-from-di-string Title: 2375. Construct Smallest Number From DI String Problem Description: You are given a 0-indexed string pattern of length n consisting of the characters 'I' meaning increasing and 'D' meaning decreasing. A 0-indexed string num of length n + 1 is created using the following conditions:

num consists of the digits '1' to '9', where each digit is used at most once. If pattern[i] == 'I', then num[i] < num[i + 1]. If pattern[i] == 'D', then num[i] > num[i + 1].

Return the lexicographically smallest possible string num that meets the conditions. Example 1: Input: pattern = "IIIDIDDD" Output: "123549876" Explanation: At indices 0, 1, 2, and 4 we must have that num[i] < num[i+1]. At indices 3, 5, 6, and 7 we must have that num[i] > num[i+1]. Some possible values of num are "245639871", "135749862", and "123849765". It can be proven that "123549876" is the smallest possible num that meets the conditions. Note that "123414321" is not possible because the digit '1' is used more than once. Example 2: Input: pattern = "DDD" Output: "4321" Explanation: Some possible values of num are "9876", "7321", and "8742". It can be proven that "4321" is the smallest possible num that meets the conditions.

Constraints:

1 <= pattern.length <= 8 pattern consists of only the letters 'I' and 'D'.

==
Problem Number: 1498 URL: https://leetcode.com/problems/time-needed-to-rearrange-a-binary-string Title: 2380. Time Needed to Rearrange a Binary String Problem Description: You are given a binary string s. In one second, all occurrences of "01" are simultaneously replaced with "10". This process repeats until no occurrences of "01" exist. Return the number of seconds needed to complete this process. Example 1: Input: s = "0110101" Output: 4 Explanation: After one second, s becomes "1011010". After another second, s becomes "1101100". After the third second, s becomes "1110100". After the fourth second, s becomes "1111000". No occurrence of "01" exists any longer, and the process needed 4 seconds to complete, so we return 4.

Example 2: Input: s = "11100" Output: 0 Explanation: No occurrence of "01" exists in s, and the processes needed 0 seconds to complete, so we return 0.

Constraints:

1 <= s.length <= 1000 s[i] is either '0' or '1'.

Follow up: Can you solve this problem in O(n) time complexity?

==
Problem Number: 1499 URL: https://leetcode.com/problems/shifting-letters-ii Title: 2381. Shifting Letters II Problem Description: You are given a string s of lowercase English letters and a 2D integer array shifts where shifts[i] = [starti, endi, directioni]. For every i, shift the characters in s from the index starti to the index endi (inclusive) forward if directioni = 1, or shift the characters backward if directioni = 0. Shifting a character forward means replacing it with the next letter in the alphabet (wrapping around so that 'z' becomes 'a'). Similarly, shifting a character backward means replacing it with the previous letter in the alphabet (wrapping around so that 'a' becomes 'z'). Return the final string after all such shifts to s are applied. Example 1: Input: s = "abc", shifts = [[0,1,0],[1,2,1],[0,2,1]] Output: "ace" Explanation: Firstly, shift the characters from index 0 to index 1 backward. Now s = "zac". Secondly, shift the characters from index 1 to index 2 forward. Now s = "zbd". Finally, shift the characters from index 0 to index 2 forward. Now s = "ace". Example 2: Input: s = "dztz", shifts = [[0,0,0],[1,1,1]] Output: "catz" Explanation: Firstly, shift the characters from index 0 to index 0 backward. Now s = "cztz". Finally, shift the characters from index 1 to index 1 forward. Now s = "catz".

Constraints:

1 <= s.length, shifts.length <= 5 * 104 shifts[i].length == 3 0 <= starti <= endi < s.length 0 <= directioni <= 1 s consists of lowercase English letters.

==
Problem Number: 1500 URL: https://leetcode.com/problems/largest-palindromic-number Title: 2384. Largest Palindromic Number Problem

Description: You are given a string num consisting of digits only. Return the largest palindromic integer (in the form of a string) that can be formed using digits taken from num. It should not contain leading zeroes. Notes:

You do not need to use all the digits of num, but you must use at least one digit. The digits can be reordered.

Example 1: Input: num = ”444947137” Output: ”7449447” Explanation: Use the digits ”4449477” from ”444947137” to form the palindromic integer ”7449447”. It can be shown that ”7449447” is the largest palindromic integer that can be formed.

Example 2: Input: num = ”00009” Output: ”9” Explanation: It can be shown that ”9” is the largest palindromic integer that can be formed. Note that the integer returned should not contain leading zeroes.

Constraints:

1 <= num.length <= 105 num consists of digits.

==
Problem Number: 1501 URL: https://leetcode.com/problems/amount-of-time-for-binary-tree-to-be-infected Title: 2385. Amount of Time for Binary Tree to Be Infected Problem Description: You are given the root of a binary tree with unique values, and an integer start. At minute 0, an infection starts from the node with value start. Each minute, a node becomes infected if:

The node is currently uninfected. The node is adjacent to an infected node.

Return the number of minutes needed for the entire tree to be infected. Example 1:

Input: root = [1,5,3,null,4,10,6,9,2], start = 3 Output: 4 Explanation: The following nodes are infected during: - Minute 0: Node 3 - Minute 1: Nodes 1, 10 and 6 - Minute 2: Node 5 - Minute 3: Node 4 - Minute 4: Nodes 9 and 2 It takes 4 minutes for the whole tree to be infected so we return 4.

Example 2:

Input: root = [1], start = 1 Output: 0 Explanation: At minute 0, the only node in the tree is infected so we return 0.

Constraints:

The number of nodes in the tree is in the range [1, 105]. 1 <= Node.val <= 105 Each node has a unique value. A node with a value of start exists in the tree.

==
Problem Number: 1502 URL: https://leetcode.com/problems/removing-stars-from-a-string Title: 2390. Removing Stars From a String Problem Description: You are given a string s, which contains stars *. In one operation, you can:

Choose a star in s. Remove the closest non-star character to its left, as well as remove the star itself.

Return the string after all stars have been removed. Note:

The input will be generated such that the operation is always possible. It can be shown that the resulting string will always be unique.

Example 1: Input: s = "leet**cod*e" Output: "lecoe" Explanation: Performing the removals from left to right: - The closest character to the 1st star is 't' in "leet**cod*e". s becomes "lee*cod*e". - The closest character to the 2nd star is 'e' in "lee*cod*e". s becomes "lecod*e". - The closest character to the 3rd star is 'd' in "lecod*e". s becomes "lecoe". There are no more stars, so we return "lecoe". Example 2: Input: s = "erase*****" Output: "" Explanation: The entire string is removed, so we return an empty string.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters and stars *. The operation above can be performed on s.

==
Problem Number: 1503 URL: https://leetcode.com/problems/minimum-amount-of-time-to-collect-garbage Title: 2391. Minimum Amount of Time to Collect Garbage Problem Description: You are given a 0-indexed array of strings garbage where garbage[i] represents the assortment of garbage at the ith house. garbage[i] consists only of the characters 'M', 'P' and 'G' representing one unit of metal, paper and glass garbage respectively. Picking up one unit of any type of garbage takes 1 minute. You are also given a 0-indexed integer array travel where travel[i] is the number of minutes needed to go from house i to house i + 1. There are three garbage trucks in the city, each responsible for picking up one type of garbage. Each garbage truck starts at house 0 and must visit each house in order; however, they do not need to visit every house. Only one garbage truck may be used at any given moment. While one truck is driving or picking up garbage, the other two trucks cannot do anything. Return the minimum number of minutes needed to pick up all the garbage. Example 1: Input: garbage = ["G","P","GP","GG"], travel = [2,4,3] Output: 21 Explanation: The paper garbage truck: 1. Travels from house 0 to house 1 2. Collects the paper garbage at house 1 3. Travels from house 1 to house 2 4. Collects the paper garbage at house 2 Altogether, it takes 8 minutes to pick up all the paper garbage. The glass garbage truck: 1. Collects the glass garbage at house 0 2. Travels from house 0 to house 1 3. Travels from house 1 to house 2 4. Collects the glass garbage at house 2 5. Travels from house 2 to house 3 6. Collects the glass garbage at house 3 Altogether, it takes 13 minutes to pick up all the glass garbage. Since there is no metal garbage, we do not need to consider the metal garbage truck. Therefore, it takes a total of 8 + 13 = 21 minutes to collect all the garbage.

Example 2: Input: garbage = ["MMM","PGM","GP"], travel = [3,10] Output:

37 Explanation: The metal garbage truck takes 7 minutes to pick up all the metal garbage. The paper garbage truck takes 15 minutes to pick up all the paper garbage. The glass garbage truck takes 15 minutes to pick up all the glass garbage. It takes a total of 7 + 15 + 15 = 37 minutes to collect all the garbage.

Constraints:

2 <= garbage.length <= 105 garbage[i] consists of only the letters 'M', 'P', and 'G'. 1 <= garbage[i].length <= 10 travel.length == garbage.length - 1 1 <= travel[i] <= 100

==

Problem Number: 1504 URL: https://leetcode.com/problems/strictly-palindromic-number Title: 2396. Strictly Palindromic Number Problem Description: An integer n is strictly palindromic if, for every base b between 2 and n - 2 (inclusive), the string representation of the integer n in base b is palindromic. Given an integer n, return true if n is strictly palindromic and false otherwise. A string is palindromic if it reads the same forward and backward. Example 1: Input: n = 9 Output: false Explanation: In base 2: 9 = 1001 (base 2), which is palindromic. In base 3: 9 = 100 (base 3), which is not palindromic. Therefore, 9 is not strictly palindromic so we return false. Note that in bases 4, 5, 6, and 7, n = 9 is also not palindromic.

Example 2: Input: n = 4 Output: false Explanation: We only consider base 2: 4 = 100 (base 2), which is not palindromic. Therefore, we return false.

Constraints:

4 <= n <= 105

==

Problem Number: 1505 URL: https://leetcode.com/problems/maximum-rows-covered-by-columns Title: 2397. Maximum Rows Covered by Columns Problem Description: You are given a 0-indexed m x n binary matrix matrix and an integer numSelect, which denotes the number of distinct columns you must select from matrix. Let us consider s = {c1, c2, ..., cnumSelect} as the set of columns selected by you. A row row is covered by s if:

For each cell matrix[row][col] (0 <= col <= n - 1) where matrix[row][col] == 1, col is present in s or, No cell in row has a value of 1.

You need to choose numSelect columns such that the number of rows that are covered is maximized. Return the maximum number of rows that can be covered by a set of numSelect columns. Example 1:

Input: matrix = [[0,0,0],[1,0,1],[0,1,1],[0,0,1]], numSelect = 2 Output: 3 Explanation: One possible way to cover 3 rows is shown in the diagram above. We choose s = {0, 2}. - Row 0 is covered because it has no occurrences of 1. - Row 1 is covered because the columns with value 1, i.e. 0 and 2 are present in s. - Row 2 is not covered because matrix[2][1] == 1 but 1 is not present in s. - Row 3 is covered because matrix[2][2] == 1 and 2 is present in s. Thus, we can cover three rows. Note that s = {1, 2} will also cover 3 rows, but it can be shown that no more than three rows can be covered.

Example 2:

Input: matrix = [[1],[0]], numSelect = 1 Output: 2 Explanation: Selecting the only column will result in both rows being covered since the entire matrix is selected. Therefore, we return 2.

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 12 matrix[i][j] is either 0 or 1. 1 <= numSelect <= n

==
Problem Number: 1506 URL: https://leetcode.com/problems/number-of-ways-to-reach-a-position-after-exactly-k-steps Title: 2400. Number of Ways to Reach a Position After Exactly k Steps Problem Description: You are given two positive integers startPos and endPos. Initially, you are standing at position startPos on an infinite number line. With one step, you can move either one position to the left, or one position to the right. Given a positive integer k, return the number of different ways to reach the position endPos starting from startPos, such that you perform exactly k steps. Since the answer may be very large, return it modulo 109 + 7. Two ways are considered different if the order of the steps made is not exactly the same. Note that the number line includes negative integers. Example 1: Input: startPos = 1, endPos = 2, k = 3 Output: 3 Explanation: We can reach position 2 from 1 in exactly 3 steps in three ways: - 1 -> 2 -> 3 -> 2. - 1 -> 2 -> 1 -> 2. - 1 -> 0 -> 1 -> 2. It can be proven that no other way is possible, so we return 3. Example 2: Input: startPos = 2, endPos = 5, k = 10 Output: 0 Explanation: It is impossible to reach position 5 from position 2 in exactly 10 steps.

Constraints:

1 <= startPos, endPos, k <= 1000

==
Problem Number: 1507 URL: https://leetcode.com/problems/longest-nice-subarray Title: 2401. Longest Nice Subarray Problem Description: You are given an array nums consisting of positive integers. We call a subarray of nums nice if the bitwise AND of every pair of elements that are in different positions in the subarray is equal to 0. Return the length of the longest nice subarray. A subarray is a contiguous part of an array. Note that subarrays of length 1 are always considered nice. Example 1: Input: nums = [1,3,8,48,10] Output: 3 Explanation: The longest nice subarray is [3,8,48]. This subarray satisfies the conditions: - 3 AND 8 = 0. - 3 AND 48 = 0. - 8 AND 48 = 0. It can be proven that no longer nice subarray can be obtained, so we return 3. Example 2: Input: nums = [3,1,5,11,13] Output: 1 Explanation: The length of the longest nice subarray is 1. Any subarray of length 1 can be chosen.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1508 URL: https://leetcode.com/problems/optimal-partition-of-string Title: 2405. Optimal Partition of String Problem Description: Given a string s, partition the string into one or more substrings such that the characters in each substring are unique. That is, no letter appears in a single substring more than once. Return the minimum number of substrings in such a partition. Note that each character should belong to exactly one substring in a partition. Example 1: Input: s = ”abacaba” Output: 4 Explanation: Two possible partitions are (”a”,”ba”,”cab”,”a”) and (”ab”,”a”,”ca”,”ba”). It can be shown that 4 is the minimum number of substrings needed.

Example 2: Input: s = ”ssssss” Output: 6 Explanation: The only valid partition is (”s”,”s”,”s”,”s”,”s”,”s”).

Constraints:

1 <= s.length <= 105 s consists of only English lowercase letters.

==
Problem Number: 1509 URL: https://leetcode.com/problems/divide-intervals-into-minimum-number-of-groups Title: 2406. Divide Intervals Into Minimum Number of Groups Problem Description: You are given a 2D integer array intervals where intervals[i] = [lefti, righti] represents the inclusive interval [lefti, righti]. You have to divide the intervals into one or more groups such that each interval is in exactly one group, and no two intervals that are in the same group intersect each other. Return the minimum number of groups you need to make. Two intervals intersect if there is at least one common number between them. For example, the intervals [1, 5] and [5, 8] intersect. Example 1: Input: intervals = [[5,10],[6,8],[1,5],[2,3],[1,10]] Output: 3 Explanation: We can divide the intervals into the following groups: - Group 1: [1, 5], [6, 8]. - Group 2: [2, 3], [5, 10]. - Group 3: [1, 10]. It can be proven that it is not possible to divide the intervals into fewer than 3 groups.

Example 2: Input: intervals = [[1,3],[5,6],[8,10],[11,13]] Output: 1 Explanation: None of the intervals overlap, so we can put all of them in one group.

Constraints:

1 <= intervals.length <= 105 intervals[i].length == 2 1 <= lefti <= righti <= 106

==
Problem Number: 1510 URL: https://leetcode.com/problems/maximum-matching-of-players-with-trainers Title: 2410. Maximum Matching of Players With Trainers Problem Description: You are given a 0-indexed integer array players, where players[i] represents the ability of the ith player. You are also given a 0-indexed integer array trainers, where trainers[j] represents the

training capacity of the jth trainer. The ith player can match with the jth trainer if the player's ability is less than or equal to the trainer's training capacity. Additionally, the ith player can be matched with at most one trainer, and the jth trainer can be matched with at most one player. Return the maximum number of matchings between players and trainers that satisfy these conditions. Example 1: Input: players = [4,7,9], trainers = [8,2,5,8] Output: 2 Explanation: One of the ways we can form two matchings is as follows: - players[0] can be matched with trainers[0] since 4 <= 8. - players[1] can be matched with trainers[3] since 7 <= 8. It can be proven that 2 is the maximum number of matchings that can be formed.

Example 2: Input: players = [1,1,1], trainers = [10] Output: 1 Explanation: The trainer can be matched with any of the 3 players. Each player can only be matched with one trainer, so the maximum answer is 1.

Constraints:

1 <= players.length, trainers.length <= 105 1 <= players[i], trainers[j] <= 109

==
Problem Number: 1511 URL: https://leetcode.com/problems/smallest-subarrays-with-maximum-bitwise-or Title: 2411. Smallest Subarrays With Maximum Bitwise OR Problem Description: You are given a 0-indexed array nums of length n, consisting of non-negative integers. For each index i from 0 to n - 1, you must determine the size of the minimum sized non-empty subarray of nums starting at i (inclusive) that has the maximum possible bitwise OR.

In other words, let Bij be the bitwise OR of the subarray nums[i...j]. You need to find the smallest subarray starting at i, such that bitwise OR of this subarray is equal to max(Bik) where i <= k <= n - 1.

The bitwise OR of an array is the bitwise OR of all the numbers in it. Return an integer array answer of size n where answer[i] is the length of the minimum sized subarray starting at i with maximum bitwise OR. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,0,2,1,3] Output: [3,3,2,2,1] Explanation: The maximum possible bitwise OR starting at any index is 3. - Starting at index 0, the shortest subarray that yields it is [1,0,2]. - Starting at index 1, the shortest subarray that yields the maximum bitwise OR is [0,2,1]. - Starting at index 2, the shortest subarray that yields the maximum bitwise OR is [2,1]. - Starting at index 3, the shortest subarray that yields the maximum bitwise OR is [1,3]. - Starting at index 4, the shortest subarray that yields the maximum bitwise OR is [3]. Therefore, we return [3,3,2,2,1].

Example 2: Input: nums = [1,2] Output: [2,1] Explanation: Starting at index 0, the shortest subarray that yields the maximum bitwise OR is of length 2. Starting at index 1, the shortest subarray that yields the maximum bitwise OR is of length 1. Therefore, we return [2,1].

Constraints:

n == nums.length 1 <= n <= 105 0 <= nums[i] <= 109

==

Problem Number: 1512 URL: https://leetcode.com/problems/length-of-the-longest-alphabetical-continuous-substring Title: 2414. Length of the Longest Alphabetical Continuous Substring Problem Description: An alphabetical continuous string is a string consisting of consecutive letters in the alphabet. In other words, it is any substring of the string ”abcdefghijklmnopqrstuvwxyz”.

For example, ”abc” is an alphabetical continuous string, while ”acb” and ”za” are not.

Given a string s consisting of lowercase letters only, return the length of the longest alphabetical continuous substring. Example 1: Input: s = ”abacaba” Output: 2 Explanation: There are 4 distinct continuous substrings: ”a”, ”b”, ”c” and ”ab”. ”ab” is the longest continuous substring.

Example 2: Input: s = ”abcde” Output: 5 Explanation: ”abcde” is the longest continuous substring.

Constraints:

1 <= s.length <= 105 s consists of only English lowercase letters.

==

Problem Number: 1513 URL: https://leetcode.com/problems/reverse-odd-levels-of-binary-tree Title: 2415. Reverse Odd Levels of Binary Tree Problem Description: Given the root of a perfect binary tree, reverse the node values at each odd level of the tree.

For example, suppose the node values at level 3 are [2,1,3,4,7,11,29,18], then it should become [18,29,11,7,4,3,1,2].

Return the root of the reversed tree. A binary tree is perfect if all parent nodes have two children and all leaves are on the same level. The level of a node is the number of edges along the path between it and the root node. Example 1:

Input: root = [2,3,5,8,13,21,34] Output: [2,5,3,8,13,21,34] Explanation: The tree has only one odd level. The nodes at level 1 are 3, 5 respectively, which are reversed and become 5, 3.

Example 2:

Input: root = [7,13,11] Output: [7,11,13] Explanation: The nodes at level 1 are 13, 11, which are reversed and become 11, 13.

Example 3: Input: root = [0,1,2,0,0,0,0,1,1,1,1,2,2,2,2] Output: [0,2,1,0,0,0,0,2,2,2,2,1,1,1,1] Explanation: The odd levels have non-zero values. The nodes at level 1 were 1, 2, and are 2, 1 after the reversal. The nodes at level 3 were 1, 1, 1, 1, 2, 2, 2, 2, and are 2, 2, 2, 2, 1, 1, 1, 1 after the reversal.

Constraints:

The number of nodes in the tree is in the range [1, 214]. 0 <= Node.val <= 105 root is a perfect binary tree.

==

Problem Number: 1514 URL: https://leetcode.com/problems/longest-subarray-with-maximum-bitwise-and Title: 2419. Longest Subarray With Maximum Bitwise AND Problem Description: You are given an integer array nums of size n. Consider a non-empty subarray from nums that has the maximum possible bitwise AND.

In other words, let k be the maximum value of the bitwise AND of any subarray of nums. Then, only subarrays with a bitwise AND equal to k should be considered.

Return the length of the longest such subarray. The bitwise AND of an array is the bitwise AND of all the numbers in it. A subarray is a contiguous sequence of elements within an array.   Example 1: Input: nums = [1,2,3,3,2,2] Output: 2 Explanation: The maximum possible bitwise AND of a subarray is 3. The longest subarray with that value is [3,3], so we return 2.

Example 2: Input: nums = [1,2,3,4] Output: 1 Explanation: The maximum possible bitwise AND of a subarray is 4. The longest subarray with that value is [4], so we return 1.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 106

==

Problem Number: 1515 URL: https://leetcode.com/problems/find-all-good-indices Title: 2420. Find All Good Indices Problem Description: You are given a 0-indexed integer array nums of size n and a positive integer k. We call an index i in the range k <= i < n - k good if the following conditions are satisfied:

The k elements that are just before the index i are in non-increasing order. The k elements that are just after the index i are in non-decreasing order.

Return an array of all good indices sorted in increasing order.   Example 1: Input: nums = [2,1,1,1,3,4,1], k = 2 Output: [2,3] Explanation: There are two good indices in the array: - Index 2. The subarray [2,1] is in non-increasing order, and the subarray [1,3] is in non-decreasing order. - Index 3. The subarray [1,1] is in non-increasing order, and the subarray [3,4] is in non-decreasing order. Note that the index 4 is not good because [4,1] is not non-decreasing. Example 2: Input: nums = [2,1,1,2], k = 2 Output: [] Explanation: There are no good indices in this array.

Constraints:

n == nums.length 3 <= n <= 105 1 <= nums[i] <= 106 1 <= k <= n / 2

==

Problem Number: 1516 URL: https://leetcode.com/problems/longest-uploaded-prefix Title: 2424. Longest Uploaded Prefix Problem Description: You are given a stream of n videos, each represented by a distinct number from 1 to n that you need to "upload" to a server. You need to implement a data structure that calculates the length of the longest uploaded prefix at various points in the upload process. We consider i to be an uploaded prefix if all videos in the range 1 to i (inclusive) have been uploaded to the server. The longest uploaded prefix is the maximum value of i that satisfies this definition.

Implement the LUPrefix class:

LUPrefix(int n) Initializes the object for a stream of n videos. void upload(int video) Uploads video to the server. int longest() Returns the length of the longest uploaded prefix defined above.

Example 1: Input ["LUPrefix", "upload", "longest", "upload", "longest", "upload", "longest"] [[4], [3], [], [1], [], [2], []] Output [null, null, 0, null, 1, null, 3]

Explanation LUPrefix server = new LUPrefix(4); // Initialize a stream of 4 videos. server.upload(3); // Upload video 3. server.longest(); // Since video 1 has not been uploaded yet, there is no prefix. // So, we return 0. server.upload(1); // Upload video 1. server.longest(); // The prefix [1] is the longest uploaded prefix, so we return 1. server.upload(2); // Upload video 2. server.longest(); // The prefix [1,2,3] is the longest uploaded prefix, so we return 3.

Constraints:

1 <= n <= 105 1 <= video <= n All values of video are distinct. At most 2 * 105 calls in total will be made to upload and longest. At least one call will be made to longest.

==

Problem Number: 1517 URL: https://leetcode.com/problems/bitwise-xor-of-all-pairings Title: 2425. Bitwise XOR of All Pairings Problem Description: You are given two 0-indexed arrays, nums1 and nums2, consisting of non-negative integers. There exists another array, nums3, which contains the bitwise XOR of all pairings of integers between nums1 and nums2 (every integer in nums1 is paired with every integer in nums2 exactly once). Return the bitwise XOR of all integers in nums3. Example 1: Input: nums1 = [2,1,3], nums2 = [10,2,5,0] Output: 13 Explanation: A possible nums3 array is [8,0,7,2,11,3,4,1,9,1,6,3]. The bitwise XOR of all these numbers is 13, so we return 13.

Example 2: Input: nums1 = [1,2], nums2 = [3,4] Output: 0 Explanation: All possible pairs of bitwise XORs are nums1[0] ^ nums2[0], nums1[0] ^ nums2[1], nums1[1] ^ nums2[0], and nums1[1] ^ nums2[1]. Thus, one possible nums3 array is [2,5,1,6]. 2 ^ 5 ^ 1 ^ 6 = 0, so we return 0.

Constraints:

1 <= nums1.length, nums2.length <= 105 0 <= nums1[i], nums2[j] <= 109

==
Problem Number: 1518 URL: https://leetcode.com/problems/maximum-sum-of-an-hourglass Title: 2428. Maximum Sum of an Hourglass Problem Description: You are given an m x n integer matrix grid. We define an hourglass as a part of the matrix with the following form:

Return the maximum sum of the elements of an hourglass. Note that an hourglass cannot be rotated and must be entirely contained within the matrix. Example 1:

Input: grid = [[6,2,1,3],[4,2,1,5],[9,2,8,7],[4,1,2,9]] Output: 30 Explanation: The cells shown above represent the hourglass with the maximum sum: 6 + 2 + 1 + 2 + 9 + 2 + 8 = 30.

Example 2:

Input: grid = [[1,2,3],[4,5,6],[7,8,9]] Output: 35 Explanation: There is only one hourglass in the matrix, with the sum: 1 + 2 + 3 + 5 + 7 + 8 + 9 = 35.

Constraints:

m == grid.length n == grid[i].length 3 <= m, n <= 150 0 <= grid[i][j] <= 106
==
Problem Number: 1519 URL: https://leetcode.com/problems/minimize-xor Title: 2429. Minimize XOR Problem Description: Given two positive integers num1 and num2, find the positive integer x such that:

x has the same number of set bits as num2, and The value x XOR num1 is minimal.

Note that XOR is the bitwise XOR operation. Return the integer x. The test cases are generated such that x is uniquely determined. The number of set bits of an integer is the number of 1's in its binary representation. Example 1: Input: num1 = 3, num2 = 5 Output: 3 Explanation: The binary representations of num1 and num2 are 0011 and 0101, respectively. The integer 3 has the same number of set bits as num2, and the value 3 XOR 3 = 0 is minimal.

Example 2: Input: num1 = 1, num2 = 12 Output: 3 Explanation: The binary representations of num1 and num2 are 0001 and 1100, respectively. The integer 3 has the same number of set bits as num2, and the value 3 XOR 1 = 2 is minimal.

Constraints:

1 <= num1, num2 <= 109
==
Problem Number: 1520 URL: https://leetcode.com/problems/find-the-original-array-of-prefix-xor Title: 2433. Find The Original Array of Prefix Xor Problem

Description: You are given an integer array pref of size n. Find and return the array arr of size n that satisfies:

pref[i] = arr[0] ^ arr[1] ^ ... ^ arr[i].

Note that ^ denotes the bitwise-xor operation. It can be proven that the answer is unique. Example 1: Input: pref = [5,2,0,3,1] Output: [5,7,2,3,2] Explanation: From the array [5,7,2,3,2] we have the following: - pref[0] = 5. - pref[1] = 5 ^ 7 = 2. - pref[2] = 5 ^ 7 ^ 2 = 0. - pref[3] = 5 ^ 7 ^ 2 ^ 3 = 3. - pref[4] = 5 ^ 7 ^ 2 ^ 3 ^ 2 = 1.

Example 2: Input: pref = [13] Output: [13] Explanation: We have pref[0] = arr[0] = 13.

Constraints:

1 <= pref.length <= 105 0 <= pref[i] <= 106

==
Problem Number: 1521 URL: https://leetcode.com/problems/using-a-robot-to-print-the-lexicographically-smallest-string Title: 2434. Using a Robot to Print the Lexicographically Smallest String Problem Description: You are given a string s and a robot that currently holds an empty string t. Apply one of the following operations until s and t are both empty:

Remove the first character of a string s and give it to the robot. The robot will append this character to the string t. Remove the last character of a string t and give it to the robot. The robot will write this character on paper.

Return the lexicographically smallest string that can be written on the paper. Example 1: Input: s = "zza" Output: "azz" Explanation: Let p denote the written string. Initially p="", s="zza", t="". Perform first operation three times p="", s="", t="zza". Perform second operation three times p="azz", s="", t="".

Example 2: Input: s = "bac" Output: "abc" Explanation: Let p denote the written string. Perform first operation twice p="", s="c", t="ba". Perform second operation twice p="ab", s="c", t="". Perform first operation p="ab", s="", t="c". Perform second operation p="abc", s="", t="".

Example 3: Input: s = "bdda" Output: "addb" Explanation: Let p denote the written string. Initially p="", s="bdda", t="". Perform first operation four times p="", s="", t="bdda". Perform second operation four times p="addb", s="", t="".

Constraints:

1 <= s.length <= 105 s consists of only English lowercase letters.

==
Problem Number: 1522 URL: https://leetcode.com/problems/range-product-queries-of-powers Title: 2438. Range Product Queries of Powers Problem

Description: Given a positive integer n, there exists a 0-indexed array called powers, composed of the minimum number of powers of 2 that sum to n. The array is sorted in non-decreasing order, and there is only one way to form the array. You are also given a 0-indexed 2D integer array queries, where queries[i] = [lefti, righti]. Each queries[i] represents a query where you have to find the product of all powers[j] with lefti <= j <= righti. Return an array answers, equal in length to queries, where answers[i] is the answer to the ith query. Since the answer to the ith query may be too large, each answers[i] should be returned modulo 109 + 7. Example 1: Input: n = 15, queries = [[0,1],[2,2],[0,3]] Output: [2,4,64] Explanation: For n = 15, powers = [1,2,4,8]. It can be shown that powers cannot be a smaller size. Answer to 1st query: powers[0] * powers[1] = 1 * 2 = 2. Answer to 2nd query: powers[2] = 4. Answer to 3rd query: powers[0] * powers[1] * powers[2] * powers[3] = 1 * 2 * 4 * 8 = 64. Each answer modulo 109 + 7 yields the same answer, so [2,4,64] is returned.

Example 2: Input: n = 2, queries = [[0,0]] Output: [2] Explanation: For n = 2, powers = [2]. The answer to the only query is powers[0] = 2. The answer modulo 109 + 7 is the same, so [2] is returned.

Constraints:

1 <= n <= 109 1 <= queries.length <= 105 0 <= starti <= endi < powers.length

==

Problem Number: 1523 URL: https://leetcode.com/problems/minimize-maximum-of-array Title: 2439. Minimize Maximum of Array Problem Description: You are given a 0-indexed array nums comprising of n non-negative integers. In one operation, you must:

Choose an integer i such that 1 <= i < n and nums[i] > 0. Decrease nums[i] by 1. Increase nums[i - 1] by 1.

Return the minimum possible value of the maximum integer of nums after performing any number of operations. Example 1: Input: nums = [3,7,1,6] Output: 5 Explanation: One set of optimal operations is as follows: 1. Choose i = 1, and nums becomes [4,6,1,6]. 2. Choose i = 3, and nums becomes [4,6,2,5]. 3. Choose i = 1, and nums becomes [5,5,2,5]. The maximum integer of nums is 5. It can be shown that the maximum number cannot be less than 5. Therefore, we return 5.

Example 2: Input: nums = [10,1] Output: 10 Explanation: It is optimal to leave nums as is, and since 10 is the maximum value, we return 10.

Constraints:

n == nums.length 2 <= n <= 105 0 <= nums[i] <= 109

==

Problem Number: 1524 URL: https://leetcode.com/problems/count-number-of-distinct-integers-after-reverse-operations Title: 2442. Count Number of Distinct Integers After Reverse Operations Problem Description: You are given an array nums consisting of positive integers. You have to take each integer in the array, reverse its digits, and add it to the end of the array. You should apply this operation to the original integers in nums. Return the number of distinct integers in the final array. Example 1: Input: nums = [1,13,10,12,31] Output: 6 Explanation: After including the reverse of each number, the resulting array is [1,13,10,12,31,1,31,1,21,13]. The reversed integers that were added to the end of the array are underlined. Note that for the integer 10, after reversing it, it becomes 01 which is just 1. The number of distinct integers in this array is 6 (The numbers 1, 10, 12, 13, 21, and 31). Example 2: Input: nums = [2,2,2] Output: 1 Explanation: After including the reverse of each number, the resulting array is [2,2,2,2,2,2]. The number of distinct integers in this array is 1 (The number 2).

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 106

==
Problem Number: 1525 URL: https://leetcode.com/problems/sum-of-number-and-its-reverse Title: 2443. Sum of Number and Its Reverse Problem Description: Given a non-negative integer num, return true if num can be expressed as the sum of any non-negative integer and its reverse, or false otherwise. Example 1: Input: num = 443 Output: true Explanation: 172 + 271 = 443 so we return true.

Example 2: Input: num = 63 Output: false Explanation: 63 cannot be expressed as the sum of a non-negative integer and its reverse so we return false.

Example 3: Input: num = 181 Output: true Explanation: 140 + 041 = 181 so we return true. Note that when a number is reversed, there may be leading zeros.

Constraints:

0 <= num <= 105

==
Problem Number: 1526 URL: https://leetcode.com/problems/number-of-subarrays-with-gcd-equal-to-k Title: 2447. Number of Subarrays With GCD Equal to K Problem Description: Given an integer array nums and an integer k, return the number of subarrays of nums where the greatest common divisor of the subarray's elements is k. A subarray is a contiguous non-empty sequence of elements within an array. The greatest common divisor of an array is the largest integer that evenly divides all the array elements. Example 1: Input: nums = [9,3,1,2,6,3], k = 3 Output: 4 Explanation: The subarrays of nums where 3 is the greatest common divisor of all the subarray's elements are: - [9,3,1,2,6,3] - [9,3,1,2,6,3] - [9,3,1,2,6,3] - [9,3,1,2,6,3]

Example 2: Input: nums = [4], k = 7 Output: 0 Explanation: There are no subarrays of nums where 7 is the greatest common divisor of all the subarray's elements.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i], k <= 109

==

Problem Number: 1527 URL: https://leetcode.com/problems/words-within-two-edits-of-dictionary Title: 2452. Words Within Two Edits of Dictionary Problem Description: You are given two string arrays, queries and dictionary. All words in each array comprise of lowercase English letters and have the same length. In one edit you can take a word from queries, and change any letter in it to any other letter. Find all words from queries that, after a maximum of two edits, equal some word from dictionary. Return a list of all words from queries, that match with some word from dictionary after a maximum of two edits. Return the words in the same order they appear in queries. Example 1: Input: queries = ["word","note","ants","wood"], dictionary = ["wood","joke","moat"] Output: ["word","note","wood"] Explanation: - Changing the 'r' in "word" to 'o' allows it to equal the dictionary word "wood". - Changing the 'n' to 'j' and the 't' to 'k' in "note" changes it to "joke". - It would take more than 2 edits for "ants" to equal a dictionary word. - "wood" can remain unchanged (0 edits) and match the corresponding dictionary word. Thus, we return ["word","note","wood"].

Example 2: Input: queries = ["yes"], dictionary = ["not"] Output: [] Explanation: Applying any two edits to "yes" cannot make it equal to "not". Thus, we return an empty array.

Constraints:

1 <= queries.length, dictionary.length <= 100 n == queries[i].length == dictionary[j].length 1 <= n <= 100 All queries[i] and dictionary[j] are composed of lowercase English letters.

==

Problem Number: 1528 URL: https://leetcode.com/problems/destroy-sequential-targets Title: 2453. Destroy Sequential Targets Problem Description: You are given a 0-indexed array nums consisting of positive integers, representing targets on a number line. You are also given an integer space. You have a machine which can destroy targets. Seeding the machine with some nums[i] allows it to destroy all targets with values that can be represented as nums[i] + c * space, where c is any non-negative integer. You want to destroy the maximum number of targets in nums. Return the minimum value of nums[i] you can seed the machine with to destroy the maximum number of targets. Example 1: Input: nums = [3,7,8,1,1,5], space = 2 Output: 1 Explanation: If we seed the machine with nums[3], then we destroy all targets equal to 1,3,5,7,9,... In this case, we would destroy 5 total targets (all except for nums[2]). It is impossible to destroy more than 5 targets, so we return nums[3].

Example 2: Input: nums = [1,3,5,2,4,6], space = 2 Output: 1 Explanation: Seeding the machine with nums[0], or nums[3] destroys 3 targets. It is not possible to destroy more than 3 targets. Since nums[0] is the minimal integer that can destroy 3 targets, we return 1.

Example 3: Input: nums = [6,2,5], space = 100 Output: 2 Explanation: Whatever initial seed we select, we can only destroy 1 target. The minimal seed is nums[1].

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= space <= 109

==
Problem Number: 1529 URL: https://leetcode.com/problems/most-popular-video-creator Title: 2456. Most Popular Video Creator Problem Description: You are given two string arrays creators and ids, and an integer array views, all of length n. The ith video on a platform was created by creator[i], has an id of ids[i], and has views[i] views. The popularity of a creator is the sum of the number of views on all of the creator's videos. Find the creator with the highest popularity and the id of their most viewed video.

If multiple creators have the highest popularity, find all of them. If multiple videos have the highest view count for a creator, find the lexicographically smallest id.

Return a 2D array of strings answer where answer[i] = [creatori, idi] means that creatori has the highest popularity and idi is the id of their most popular video. The answer can be returned in any order. Example 1: Input: creators = ["alice","bob","alice","chris"], ids = ["one","two","three","four"], views = [5,10,5,4] Output: [["alice","one"],["bob","two"]] Explanation: The popularity of alice is 5 + 5 = 10. The popularity of bob is 10. The popularity of chris is 4. alice and bob are the most popular creators. For bob, the video with the highest view count is "two". For alice, the videos with the highest view count are "one" and "three". Since "one" is lexicographically smaller than "three", it is included in the answer.

Example 2: Input: creators = ["alice","alice","alice"], ids = ["a","b","c"], views = [1,2,2] Output: [["alice","b"]] Explanation: The videos with id "b" and "c" have the highest view count. Since "b" is lexicographically smaller than "c", it is included in the answer.

Constraints:

n == creators.length == ids.length == views.length 1 <= n <= 105 1 <= creators[i].length, ids[i].length <= 5 creators[i] and ids[i] consist only of lowercase English letters. 0 <= views[i] <= 105

==
Problem Number: 1530 URL: https://leetcode.com/problems/minimum-addition-to-make-integer-beautiful Title: 2457. Minimum Addition to Make Integer Beautiful Problem Description: You are given two positive integers n and target. An integer is considered beautiful if the sum of its digits is less than or equal to target. Return the minimum non-negative integer x such that n + x is beautiful. The input will be generated such that it is always possible to make n beautiful. Example 1: Input: n = 16, target = 6 Output: 4 Explanation: Initially n is 16 and its digit sum is 1 + 6 = 7. After adding 4, n becomes 20 and digit sum becomes 2 + 0 = 2. It can be shown that we can not make n beautiful with adding non-negative integer less than 4.

Example 2: Input: n = 467, target = 6 Output: 33 Explanation: Initially n is 467 and its digit sum is 4 + 6 + 7 = 17. After adding 33, n becomes 500 and digit sum becomes 5 + 0 + 0 = 5. It can be shown that we can not make n beautiful with adding non-negative integer less than 33.

Example 3: Input: n = 1, target = 1 Output: 0 Explanation: Initially n is 1 and its digit sum is 1, which is already smaller than or equal to target.

Constraints:

1 <= n <= 1012 1 <= target <= 150 The input will be generated such that it is always possible to make n beautiful.

==
Problem Number: 1531 URL: https://leetcode.com/problems/maximum-sum-of-distinct-subarrays-with-length-k Title: 2461. Maximum Sum of Distinct Subarrays With Length K Problem Description: You are given an integer array nums and an integer k. Find the maximum subarray sum of all the subarrays of nums that meet the following conditions:

The length of the subarray is k, and All the elements of the subarray are distinct.

Return the maximum subarray sum of all the subarrays that meet the conditions. If no subarray meets the conditions, return 0. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,5,4,2,9,9,9], k = 3 Output: 15 Explanation: The subarrays of nums with length 3 are: - [1,5,4] which meets the requirements and has a sum of 10. - [5,4,2] which meets the requirements and has a sum of 11. - [4,2,9] which meets the requirements and has a sum of 15. - [2,9,9] which does not meet the requirements because the element 9 is repeated. - [9,9,9] which does not meet the requirements because the element 9 is repeated. We return 15 because it is the maximum subarray sum of all the subarrays that meet the conditions

Example 2: Input: nums = [4,4,4], k = 3 Output: 0 Explanation: The subarrays of nums with length 3 are: - [4,4,4] which does not meet the requirements because the element 4 is repeated. We return 0 because no subarrays meet the conditions.

Constraints:

1 <= k <= nums.length <= 105 1 <= nums[i] <= 105

==
Problem Number: 1532 URL: https://leetcode.com/problems/total-cost-to-hire-k-workers Title: 2462. Total Cost to Hire K Workers Problem Description: You are given a 0-indexed integer array costs where costs[i] is the cost of hiring the ith worker. You are also given two integers k and candidates. We want to hire exactly k workers according to the following rules:

You will run k sessions and hire exactly one worker in each session. In each hiring session, choose the worker with the lowest cost from either the first candidates workers or the last candidates workers. Break the tie by the smallest index.

For example, if costs = [3,2,7,7,1,2] and candidates = 2, then in the first hiring session, we will choose the 4th worker because they have the lowest cost [3,2,7,7,1,2]. In the second hiring session, we will choose 1st worker because they have the same lowest cost as 4th worker but they have the smallest index [3,2,7,7,2]. Please note that the indexing may be changed in the process.

If there are fewer than candidates workers remaining, choose the worker with the lowest cost among them. Break the tie by the smallest index. A worker can only be chosen once.

Return the total cost to hire exactly k workers. Example 1: Input: costs = [17,12,10,2,7,2,11,20,8], k = 3, candidates = 4 Output: 11 Explanation: We hire 3 workers in total. The total cost is initially 0. - In the first hiring round we choose the worker from [17,12,10,2,7,2,11,20,8]. The lowest cost is 2, and we break the tie by the smallest index, which is 3. The total cost = 0 + 2 = 2. - In the second hiring round we choose the worker from [17,12,10,7,2,11,20,8]. The lowest cost is 2 (index 4). The total cost = 2 + 2 = 4. - In the third hiring round we choose the worker from [17,12,10,7,11,20,8]. The lowest cost is 7 (index 3). The total cost = 4 + 7 = 11. Notice that the worker with index 3 was common in the first and last four workers. The total hiring cost is 11.

Example 2: Input: costs = [1,2,4,1], k = 3, candidates = 3 Output: 4 Explanation: We hire 3 workers in total. The total cost is initially 0. - In the first hiring round we choose the worker from [1,2,4,1]. The lowest cost is 1, and we break the tie by the smallest index, which is 0. The total cost = 0 + 1 = 1. Notice that workers with index 1 and 2 are common in the first and last 3 workers. - In the second hiring round we choose the worker from [2,4,1]. The lowest cost is 1 (index 2). The total cost = 1 + 1 = 2. - In the third hiring round there are less than three candidates. We choose the worker from the remaining workers [2,4]. The lowest cost is 2 (index 0). The total cost = 2 + 2 = 4. The total hiring cost is 4.

Constraints:

1 <= costs.length <= 105 1 <= costs[i] <= 105 1 <= k, candidates <=

costs.length

==

Problem Number: 1533 URL: https://leetcode.com/problems/count-ways-to-build-good-strings Title: 2466. Count Ways To Build Good Strings Problem Description: Given the integers zero, one, low, and high, we can construct a string by starting with an empty string, and then at each step perform either of the following:

Append the character '0' zero times. Append the character '1' one times.

This can be performed any number of times. A good string is a string constructed by the above process having a length between low and high (inclusive). Return the number of different good strings that can be constructed satisfying these properties. Since the answer can be large, return it modulo 109 + 7. Example 1: Input: low = 3, high = 3, zero = 1, one = 1 Output: 8 Explanation: One possible valid good string is "011". It can be constructed as follows: "" -> "0" -> "01" -> "011". All binary strings from "000" to "111" are good strings in this example.

Example 2: Input: low = 2, high = 3, zero = 1, one = 2 Output: 5 Explanation: The good strings are "00", "11", "000", "110", and "011".

Constraints:

1 <= low <= high <= 105 1 <= zero, one <= low

==

Problem Number: 1534 URL: https://leetcode.com/problems/most-profitable-path-in-a-tree Title: 2467. Most Profitable Path in a Tree Problem Description: There is an undirected tree with n nodes labeled from 0 to n - 1, rooted at node 0. You are given a 2D integer array edges of length n - 1 where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. At every node i, there is a gate. You are also given an array of even integers amount, where amount[i] represents:

the price needed to open the gate at node i, if amount[i] is negative, or, the cash reward obtained on opening the gate at node i, otherwise.

The game goes on as follows:

Initially, Alice is at node 0 and Bob is at node bob. At every second, Alice and Bob each move to an adjacent node. Alice moves towards some leaf node, while Bob moves towards node 0. For every node along their path, Alice and Bob either spend money to open the gate at that node, or accept the reward. Note that:

If the gate is already open, no price will be required, nor will there be any cash reward. If Alice and Bob reach the node simultaneously, they share the price/reward for opening the gate there. In other words, if the price to open the gate is c, then both Alice and Bob pay c / 2 each. Similarly, if the reward at the gate is c, both of them receive c / 2 each.

If Alice reaches a leaf node, she stops moving. Similarly, if Bob reaches node 0, he stops moving. Note that these events are independent of each other.

Return the maximum net income Alice can have if she travels towards the optimal leaf node.   Example 1:

Input: edges = [[0,1],[1,2],[1,3],[3,4]], bob = 3, amount = [-2,4,2,-4,6] Output: 6 Explanation: The above diagram represents the given tree. The game goes as follows: - Alice is initially on node 0, Bob on node 3. They open the gates of their respective nodes. Alice's net income is now -2. - Both Alice and Bob move to node 1.   Since they reach here simultaneously, they open the gate together and share the reward.   Alice's net income becomes -2 + (4 / 2) = 0. - Alice moves on to node 3. Since Bob already opened its gate, Alice's income remains unchanged.   Bob moves on to node 0, and stops moving. - Alice moves on to node 4 and opens the gate there. Her net income becomes 0 + 6 = 6. Now, neither Alice nor Bob can make any further moves, and the game ends. It is not possible for Alice to get a higher net income.

Example 2:

Input: edges = [[0,1]], bob = 1, amount = [-7280,2350] Output: -7280 Explanation: Alice follows the path 0->1 whereas Bob follows the path 1->0. Thus, Alice opens the gate at node 0 only. Hence, her net income is -7280.

Constraints:

2 <= n <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi edges represents a valid tree. 1 <= bob < n amount.length == n amount[i] is an even integer in the range [-104, 104].

==
Problem Number: 1535 URL: https://leetcode.com/problems/number-of-subarrays-with-lcm-equal-to-k Title: 2470. Number of Subarrays With LCM Equal to K Problem Description: Given an integer array nums and an integer k, return the number of subarrays of nums where the least common multiple of the subarray's elements is k. A subarray is a contiguous non-empty sequence of elements within an array. The least common multiple of an array is the smallest positive integer that is divisible by all the array elements.   Example 1: Input: nums = [3,6,2,7,1], k = 6 Output: 4 Explanation: The subarrays of nums where 6 is the least common multiple of all the subarray's elements are: - [3,6,2,7,1] - [3,6,2,7,1] - [3,6,2,7,1] - [3,6,2,7,1]

Example 2: Input: nums = [3], k = 2 Output: 0 Explanation: There are no subarrays of nums where 2 is the least common multiple of all the subarray's elements.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i], k <= 1000

==
Problem Number: 1536 URL: https://leetcode.com/problems/minimum-number-of-operations-to-sort-a-binary-tree-by-level Title: 2471. Minimum Number of Operations to Sort a Binary Tree by Level Problem Description: You are given the root of a binary tree with unique values. In one operation, you can choose any two nodes at the same level and swap their values. Return the minimum number of operations needed to make the values at each level sorted in a strictly increasing order. The level of a node is the number of edges along the path between it and the root node.   Example 1:

Input: root = [1,4,3,7,6,8,5,null,null,null,null,9,null,10] Output: 3 Explanation: - Swap 4 and 3. The 2nd level becomes [3,4]. - Swap 7 and 5. The 3rd level becomes [5,6,8,7]. - Swap 8 and 7. The 3rd level becomes [5,6,7,8]. We used 3 operations so return 3. It can be proven that 3 is the minimum number of operations needed.

Example 2:

Input: root = [1,3,2,7,6,5,4] Output: 3 Explanation: - Swap 3 and 2. The 2nd level becomes [2,3]. - Swap 7 and 4. The 3rd level becomes [4,6,5,7]. - Swap 6 and 5. The 3rd level becomes [4,5,6,7]. We used 3 operations so return 3. It can be proven that 3 is the minimum number of operations needed.

Example 3:

Input: root = [1,2,3,4,5,6] Output: 0 Explanation: Each level is already sorted in increasing order so return 0.

Constraints:

The number of nodes in the tree is in the range [1, 105]. 1 <= Node.val <= 105 All the values of the tree are unique.

==
Problem Number: 1537 URL: https://leetcode.com/problems/closest-nodes-queries-in-a-binary-search-tree Title: 2476. Closest Nodes Queries in a Binary Search Tree Problem Description: You are given the root of a binary search tree and an array queries of size n consisting of positive integers. Find a 2D array answer of size n where answer[i] = [mini, maxi]:

mini is the largest value in the tree that is smaller than or equal to queries[i]. If a such value does not exist, add -1 instead. maxi is the smallest value in the tree that is greater than or equal to queries[i]. If a such value does not exist, add -1 instead.

Return the array answer.   Example 1:

Input: root = [6,2,13,1,4,9,15,null,null,null,null,null,null,14], queries = [2,5,16] Output: [[2,2],[4,6],[15,-1]] Explanation: We answer the queries in the following way: - The largest number that is smaller or equal than 2 in the tree is 2, and the smallest number that is greater or equal than 2 is still 2. So the answer for the first query is [2,2]. - The largest number that is smaller or equal than 5 in the tree is 4, and the smallest number that is greater or equal than 5 is 6. So the answer for the second query is [4,6]. - The largest number that is smaller or equal than 16 in the tree is 15, and the smallest number that is greater or equal than 16 does not exist. So the answer for the third query is [15,-1].

Example 2:

Input: root = [4,null,9], queries = [3] Output: [[-1,4]] Explanation: The largest number that is smaller or equal to 3 in the tree does not exist, and the smallest number that is greater or equal to 3 is 4. So the answer for the query is [-1,4].

Constraints:

The number of nodes in the tree is in the range [2, 105]. 1 <= Node.val <= 106 n == queries.length 1 <= n <= 105 1 <= queries[i] <= 106

==

Problem Number: 1538 URL: https://leetcode.com/problems/minimum-fuel-cost-to-report-to-the-capital Title: 2477. Minimum Fuel Cost to Report to the Capital Problem Description: There is a tree (i.e., a connected, undirected graph with no cycles) structure country network consisting of n cities numbered from 0 to n - 1 and exactly n - 1 roads. The capital city is city 0. You are given a 2D integer array roads where roads[i] = [ai, bi] denotes that there exists a bidirectional road connecting cities ai and bi. There is a meeting for the representatives of each city. The meeting is in the capital city. There is a car in each city. You are given an integer seats that indicates the number of seats in each car. A representative can use the car in their city to travel or change the car and ride with another representative. The cost of traveling between two cities is one liter of fuel. Return the minimum number of liters of fuel to reach the capital city. Example 1:

Input: roads = [[0,1],[0,2],[0,3]], seats = 5 Output: 3 Explanation: - Representative1 goes directly to the capital with 1 liter of fuel. - Representative2 goes directly to the capital with 1 liter of fuel. - Representative3 goes directly to the capital with 1 liter of fuel. It costs 3 liters of fuel at minimum. It can be proven that 3 is the minimum number of liters of fuel needed.

Example 2:

Input: roads = [[3,1],[3,2],[1,0],[0,4],[0,5],[4,6]], seats = 2 Output: 7 Explanation: - Representative2 goes directly to city 3 with 1 liter of fuel. - Representative2 and representative3 go together to city 1 with 1 liter of fuel. - Representative2 and representative3 go together to the capital with 1 liter of fuel. - Representative1 goes directly to the capital with 1 liter of fuel. - Representative5 goes directly to the capital with 1 liter of fuel. - Representative6 goes directly to city 4 with 1 liter of fuel. - Representative4 and representative6 go together to the capital with 1 liter of fuel. It costs 7 liters of fuel at minimum. It can be proven that 7 is the minimum number of liters of fuel needed.

Example 3:

Input: roads = [], seats = 1 Output: 0 Explanation: No representatives need to travel to the capital city.

Constraints:

1 <= n <= 105 roads.length == n - 1 roads[i].length == 2 0 <= ai, bi < n ai != bi roads represents a valid tree. 1 <= seats <= 105

==
Problem Number: 1539 URL: https://leetcode.com/problems/difference-between-ones-and-zeros-in-row-and-column Title: 2482. Difference Between Ones and Zeros in Row and Column Problem Description: You are given a 0-indexed m x n binary matrix grid. A 0-indexed m x n difference matrix diff is created with the following procedure:

Let the number of ones in the ith row be onesRowi. Let the number of ones in the jth column be onesColj. Let the number of zeros in the ith row be zerosRowi. Let the number of zeros in the jth column be zerosColj. diff[i][j] = onesRowi + onesColj - zerosRowi - zerosColj

Return the difference matrix diff.   Example 1:

Input: grid = [[0,1,1],[1,0,1],[0,0,1]] Output: [[0,0,4],[0,0,4],[-2,-2,2]] Explanation:
- diff[0][0] = onesRow0 + onesCol0 - zerosRow0 - zerosCol0 = 2 + 1 - 1 - 2 = 0
- diff[0][1] = onesRow0 + onesCol1 - zerosRow0 - zerosCol1 = 2 + 1 - 1 - 2 = 0
- diff[0][2] = onesRow0 + onesCol2 - zerosRow0 - zerosCol2 = 2 + 3 - 1 - 0 = 4
- diff[1][0] = onesRow1 + onesCol0 - zerosRow1 - zerosCol0 = 2 + 1 - 1 - 2 = 0
- diff[1][1] = onesRow1 + onesCol1 - zerosRow1 - zerosCol1 = 2 + 1 - 1 - 2 = 0
- diff[1][2] = onesRow1 + onesCol2 - zerosRow1 - zerosCol2 = 2 + 3 - 1 - 0 = 4
- diff[2][0] = onesRow2 + onesCol0 - zerosRow2 - zerosCol0 = 1 + 1 - 2 - 2 = -2
- diff[2][1] = onesRow2 + onesCol1 - zerosRow2 - zerosCol1 = 1 + 1 - 2 - 2 = -2
- diff[2][2] = onesRow2 + onesCol2 - zerosRow2 - zerosCol2 = 1 + 3 - 2 - 0 = 2

Example 2:

Input: grid = [[1,1,1],[1,1,1]] Output: [[5,5,5],[5,5,5]] Explanation: - diff[0][0] = onesRow0 + onesCol0 - zerosRow0 - zerosCol0 = 3 + 2 - 0 - 0 = 5 - diff[0][1] = onesRow0 + onesCol1 - zerosRow0 - zerosCol1 = 3 + 2 - 0 - 0 = 5 - diff[0][2] = onesRow0 + onesCol2 - zerosRow0 - zerosCol2 = 3 + 2 - 0 - 0 = 5 - diff[1][0] = onesRow1 + onesCol0 - zerosRow1 - zerosCol0 = 3 + 2 - 0 - 0 = 5 - diff[1][1] = onesRow1 + onesCol1 - zerosRow1 - zerosCol1 = 3 + 2 - 0 - 0 = 5 - diff[1][2] = onesRow1 + onesCol2 - zerosRow1 - zerosCol2 = 3 + 2 - 0 - 0 = 5

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 1 <= m * n <= 105 grid[i][j] is either 0 or 1.

==
Problem Number: 1540 URL: https://leetcode.com/problems/minimum-

penalty-for-a-shop Title: 2483. Minimum Penalty for a Shop Problem Description: You are given the customer visit log of a shop represented by a 0-indexed string customers consisting only of characters 'N' and 'Y':

if the ith character is 'Y', it means that customers come at the ith hour whereas 'N' indicates that no customers come at the ith hour.

If the shop closes at the jth hour (0 <= j <= n), the penalty is calculated as follows:

For every hour when the shop is open and no customers come, the penalty increases by 1. For every hour when the shop is closed and customers come, the penalty increases by 1.

Return the earliest hour at which the shop must be closed to incur a minimum penalty. Note that if a shop closes at the jth hour, it means the shop is closed at the hour j. Example 1: Input: customers = "YYNY" Output: 2 Explanation: - Closing the shop at the 0th hour incurs in 1+1+0+1 = 3 penalty. - Closing the shop at the 1st hour incurs in 0+1+0+1 = 2 penalty. - Closing the shop at the 2nd hour incurs in 0+0+0+1 = 1 penalty. - Closing the shop at the 3rd hour incurs in 0+0+1+1 = 2 penalty. - Closing the shop at the 4th hour incurs in 0+0+1+0 = 1 penalty. Closing the shop at 2nd or 4th hour gives a minimum penalty. Since 2 is earlier, the optimal closing time is 2.

Example 2: Input: customers = "NNNNN" Output: 0 Explanation: It is best to close the shop at the 0th hour as no customers arrive. Example 3: Input: customers = "YYYY" Output: 4 Explanation: It is best to close the shop at the 4th hour as customers arrive at each hour.

Constraints:

1 <= customers.length <= 105 customers consists only of characters 'Y' and 'N'.

==

Problem Number: 1541 URL: https://leetcode.com/problems/append-characters-to-string-to-make-subsequence Title: 2486. Append Characters to String to Make Subsequence Problem Description: You are given two strings s and t consisting of only lowercase English letters. Return the minimum number of characters that need to be appended to the end of s so that t becomes a subsequence of s. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. Example 1: Input: s = "coaching", t = "coding" Output: 4 Explanation: Append the characters "ding" to the end of s so that s = "coachingding". Now, t is a subsequence of s ("coachingding"). It can be shown that appending any 3 characters to the end of s will never make t a subsequence.

Example 2: Input: s = "abcde", t = "a" Output: 0 Explanation: t is already a subsequence of s ("abcde").

Example 3: Input: s = ”z”, t = ”abcde” Output: 5 Explanation: Append the characters ”abcde” to the end of s so that s = ”zabcde”. Now, t is a subsequence of s (”zabcde”). It can be shown that appending any 4 characters to the end of s will never make t a subsequence.

Constraints:

1 <= s.length, t.length <= 105 s and t consist only of lowercase English letters.

==
Problem Number: 1542 URL: https://leetcode.com/problems/remove-nodes-from-linked-list Title: 2487. Remove Nodes From Linked List Problem Description: You are given the head of a linked list. Remove every node which has a node with a greater value anywhere to the right side of it. Return the head of the modified linked list. Example 1:

Input: head = [5,2,13,3,8] Output: [13,8] Explanation: The nodes that should be removed are 5, 2 and 3. - Node 13 is to the right of node 5. - Node 13 is to the right of node 2. - Node 8 is to the right of node 3.

Example 2: Input: head = [1,1,1,1] Output: [1,1,1,1] Explanation: Every node has value 1, so no nodes are removed.

Constraints:

The number of the nodes in the given list is in the range [1, 105]. 1 <= Node.val <= 105

==
Problem Number: 1543 URL: https://leetcode.com/problems/divide-players-into-teams-of-equal-skill Title: 2491. Divide Players Into Teams of Equal Skill Problem Description: You are given a positive integer array skill of even length n where skill[i] denotes the skill of the ith player. Divide the players into n / 2 teams of size 2 such that the total skill of each team is equal. The chemistry of a team is equal to the product of the skills of the players on that team. Return the sum of the chemistry of all the teams, or return -1 if there is no way to divide the players into teams such that the total skill of each team is equal. Example 1: Input: skill = [3,2,5,1,3,4] Output: 22 Explanation: Divide the players into the following teams: (1, 5), (2, 4), (3, 3), where each team has a total skill of 6. The sum of the chemistry of all the teams is: 1 * 5 + 2 * 4 + 3 * 3 = 5 + 8 + 9 = 22.

Example 2: Input: skill = [3,4] Output: 12 Explanation: The two players form a team with a total skill of 7. The chemistry of the team is 3 * 4 = 12.

Example 3: Input: skill = [1,1,2,3] Output: -1 Explanation: There is no way to divide the players into teams such that the total skill of each team is equal.

Constraints:

2 <= skill.length <= 105 skill.length is even. 1 <= skill[i] <= 1000

==

Problem Number: 1544 URL: https://leetcode.com/problems/minimum-score-of-a-path-between-two-cities Title: 2492. Minimum Score of a Path Between Two Cities Problem Description: You are given a positive integer n representing n cities numbered from 1 to n. You are also given a 2D array roads where roads[i] = [ai, bi, distancei] indicates that there is a bidirectional road between cities ai and bi with a distance equal to distancei. The cities graph is not necessarily connected. The score of a path between two cities is defined as the minimum distance of a road in this path. Return the minimum possible score of a path between cities 1 and n. Note:

A path is a sequence of roads between two cities. It is allowed for a path to contain the same road multiple times, and you can visit cities 1 and n multiple times along the path. The test cases are generated such that there is at least one path between 1 and n.

Example 1:

Input: n = 4, roads = [[1,2,9],[2,3,6],[2,4,5],[1,4,7]] Output: 5 Explanation: The path from city 1 to 4 with the minimum score is: 1 -> 2 -> 4. The score of this path is min(9,5) = 5. It can be shown that no other path has less score.

Example 2:

Input: n = 4, roads = [[1,2,2],[1,3,4],[3,4,7]] Output: 2 Explanation: The path from city 1 to 4 with the minimum score is: 1 -> 2 -> 1 -> 3 -> 4. The score of this path is min(2,2,4,7) = 2.

Constraints:

2 <= n <= 105 1 <= roads.length <= 105 roads[i].length == 3 1 <= ai, bi <= n ai != bi 1 <= distancei <= 104 There are no repeated edges. There is at least one path between 1 and n.

==

Problem Number: 1545 URL: https://leetcode.com/problems/maximum-star-sum-of-a-graph Title: 2497. Maximum Star Sum of a Graph Problem Description: There is an undirected graph consisting of n nodes numbered from 0 to n - 1. You are given a 0-indexed integer array vals of length n where vals[i] denotes the value of the ith node. You are also given a 2D integer array edges where edges[i] = [ai, bi] denotes that there exists an undirected edge connecting nodes ai and bi. A star graph is a subgraph of the given graph having a center node containing 0 or more neighbors. In other words, it is a subset of edges of the given graph such that there exists a common node for all edges. The image below shows star graphs with 3 and 4 neighbors respectively, centered at the blue node.

The star sum is the sum of the values of all the nodes present in the star graph. Given an integer k, return the maximum star sum of a star graph containing at most k edges. Example 1:

Input: vals = [1,2,3,4,10,-10,-20], edges = [[0,1],[1,2],[1,3],[3,4],[3,5],[3,6]], k = 2 Output: 16 Explanation: The above diagram represents the input graph. The star graph with the maximum star sum is denoted by blue. It is centered at 3 and includes its neighbors 1 and 4. It can be shown it is not possible to get a star graph with a sum greater than 16.

Example 2: Input: vals = [-5], edges = [], k = 0 Output: -5 Explanation: There is only one possible star graph, which is node 0 itself. Hence, we return -5.

Constraints:

n == vals.length 1 <= n <= 105 -104 <= vals[i] <= 104 0 <= edges.length <= min(n * (n - 1) / 2, 105) edges[i].length == 2 0 <= ai, bi <= n - 1 ai != bi 0 <= k <= n - 1

==
Problem Number: 1546 URL: https://leetcode.com/problems/frog-jump-ii Title: 2498. Frog Jump II Problem Description: You are given a 0-indexed integer array stones sorted in strictly increasing order representing the positions of stones in a river. A frog, initially on the first stone, wants to travel to the last stone and then return to the first stone. However, it can jump to any stone at most once. The length of a jump is the absolute difference between the position of the stone the frog is currently on and the position of the stone to which the frog jumps.

More formally, if the frog is at stones[i] and is jumping to stones[j], the length of the jump is |stones[i] - stones[j]|.

The cost of a path is the maximum length of a jump among all jumps in the path. Return the minimum cost of a path for the frog. Example 1:

Input: stones = [0,2,5,6,7] Output: 5 Explanation: The above figure represents one of the optimal paths the frog can take. The cost of this path is 5, which is the maximum length of a jump. Since it is not possible to achieve a cost of less than 5, we return it.

Example 2:

Input: stones = [0,3,9] Output: 9 Explanation: The frog can jump directly to the last stone and come back to the first stone. In this case, the length of each jump will be 9. The cost for the path will be max(9, 9) = 9. It can be shown that this is the minimum achievable cost.

Constraints:

2 <= stones.length <= 105 0 <= stones[i] <= 109 stones[0] == 0 stones is sorted in a strictly increasing order.

==
Problem Number: 1547 URL: https://leetcode.com/problems/longest-square-streak-in-an-array Title: 2501. Longest Square Streak in an Array Problem

Description: You are given an integer array nums. A subsequence of nums is called a square streak if:

The length of the subsequence is at least 2, and after sorting the subsequence, each element (except the first element) is the square of the previous number.

Return the length of the longest square streak in nums, or return -1 if there is no square streak. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.   Example 1: Input: nums = [4,3,6,16,8,2] Output: 3 Explanation: Choose the subsequence [4,16,2]. After sorting it, it becomes [2,4,16]. - 4 = 2 * 2. - 16 = 4 * 4. Therefore, [4,16,2] is a square streak. It can be shown that every subsequence of length 4 is not a square streak.

Example 2: Input: nums = [2,3,5,6,7] Output: -1 Explanation: There is no square streak in nums so return -1.

Constraints:

2 <= nums.length <= 105 2 <= nums[i] <= 105

==
Problem Number: 1548 URL: https://leetcode.com/problems/design-memory-allocator Title: 2502. Design Memory Allocator Problem Description: You are given an integer n representing the size of a 0-indexed memory array. All memory units are initially free. You have a memory allocator with the following functionalities:

Allocate a block of size consecutive free memory units and assign it the id mID. Free all memory units with the given id mID.

Note that:

Multiple blocks can be allocated to the same mID. You should free all the memory units with mID, even if they were allocated in different blocks.

Implement the Allocator class:

Allocator(int n) Initializes an Allocator object with a memory array of size n. int allocate(int size, int mID) Find the leftmost block of size consecutive free memory units and allocate it with the id mID. Return the block's first index. If such a block does not exist, return -1. int free(int mID) Free all memory units with the id mID. Return the number of memory units you have freed.

Example 1: Input ["Allocator", "allocate", "allocate", "allocate", "free", "allocate", "allocate", "allocate", "free", "allocate", "free"] [[10], [1, 1], [1, 2], [1, 3], [2], [3, 4], [1, 1], [1, 1], [1], [10, 2], [7]] Output [null, 0, 1, 2, 1, 3, 1, 6, 3, -1, 0]

Explanation Allocator loc = new Allocator(10); // Initialize a memory array of size 10. All memory units are initially free. loc.allocate(1, 1); // The leftmost block's first index is 0. The memory array becomes [1,_,_,_,_,_,_,_,_,_]. We return 0. loc.allocate(1, 2); // The leftmost block's first index is 1. The memory array becomes [1,2,_,_,_,_,_,_,_,_]. We return 1. loc.allocate(1, 3); // The leftmost block's first index is 2. The memory array becomes [1,2,3,_,_,_,_,_,_,_]. We return 2. loc.free(2); // Free all memory units with mID 2. The memory array becomes [1,_, 3,_,_,_,_,_,_,_]. We return 1 since there is only 1 unit with mID 2. loc.allocate(3, 4); // The leftmost block's first index is 3. The memory array becomes [1,_,3,4,4,4,_,_,_,_]. We return 3. loc.allocate(1, 1); // The leftmost block's first index is 1. The memory array becomes [1,1,3,4,4,4,_,_,_,_]. We return 1. loc.allocate(1, 1); // The leftmost block's first index is 6. The memory array becomes [1,1,3,4,4,4,1,_,_,_]. We return 6. loc.free(1); // Free all memory units with mID 1. The memory array becomes [_,_,3,4,4,4,_,_,_,_]. We return 3 since there are 3 units with mID 1. loc.allocate(10, 2); // We can not find any free block with 10 consecutive free memory units, so we return -1. loc.free(7); // Free all memory units with mID 7. The memory array remains the same since there is no memory unit with mID 7. We return 0.

Constraints:

1 <= n, size, mID <= 1000 At most 1000 calls will be made to allocate and free.

==
Problem Number: 1549 URL: https://leetcode.com/problems/smallest-value-after-replacing-with-sum-of-prime-factors Title: 2507. Smallest Value After Replacing With Sum of Prime Factors Problem Description: You are given a positive integer n. Continuously replace n with the sum of its prime factors.

Note that if a prime factor divides n multiple times, it should be included in the sum as many times as it divides n.

Return the smallest value n will take on. Example 1: Input: n = 15 Output: 5 Explanation: Initially, n = 15. 15 = 3 * 5, so replace n with 3 + 5 = 8. 8 = 2 * 2 * 2, so replace n with 2 + 2 + 2 = 6. 6 = 2 * 3, so replace n with 2 + 3 = 5. 5 is the smallest value n will take on.

Example 2: Input: n = 3 Output: 3 Explanation: Initially, n = 3. 3 is the smallest value n will take on.

Constraints:

2 <= n <= 105

==
Problem Number: 1550 URL: https://leetcode.com/problems/reward-top-k-students Title: 2512. Reward Top K Students Problem Description: You are given two string arrays positive_feedback and negative_feedback, containing the words denoting positive and negative feedback, respectively. Note that no word is both positive and negative. Initially every student has 0 points. Each positive word in a feedback report increases the points of a student by 3, whereas each negative word decreases the points by 1. You are given n feedback

reports, represented by a 0-indexed string array report and a 0-indexed integer array student_id, where student_id[i] represents the ID of the student who has received the feedback report report[i]. The ID of each student is unique. Given an integer k, return the top k students after ranking them in non-increasing order by their points. In case more than one student has the same points, the one with the lower ID ranks higher. Example 1: Input: positive_feedback = ["smart","brilliant","studious"], negative_feedback = ["not"], report = ["this student is studious","the student is smart"], student_id = [1,2], k = 2 Output: [1,2] Explanation: Both the students have 1 positive feedback and 3 points but since student 1 has a lower ID he ranks higher.

Example 2: Input: positive_feedback = ["smart","brilliant","studious"], negative_feedback = ["not"], report = ["this student is not studious","the student is smart"], student_id = [1,2], k = 2 Output: [2,1] Explanation: - The student with ID 1 has 1 positive feedback and 1 negative feedback, so he has 3-1=2 points. - The student with ID 2 has 1 positive feedback, so he has 3 points. Since student 2 has more points, [2,1] is returned.

Constraints:

1 <= positive_feedback.length, negative_feedback.length <= 104 1 <= positive_feedback[i].length, negative_feedback[j].length <= 100 Both positive_feedback[i] and negative_feedback[j] consists of lowercase English letters. No word is present in both positive_feedback and negative_feedback. n == report.length == student_id.length 1 <= n <= 104 report[i] consists of lowercase English letters and spaces ' '. There is a single space between consecutive words of report[i]. 1 <= report[i].length <= 100 1 <= student_id[i] <= 109 All the values of student_id[i] are unique. 1 <= k <= n

==
Problem Number: 1551 URL: https://leetcode.com/problems/minimize-the-maximum-of-two-arrays Title: 2513. Minimize the Maximum of Two Arrays Problem Description: We have two arrays arr1 and arr2 which are initially empty. You need to add positive integers to them such that they satisfy all the following conditions:

arr1 contains uniqueCnt1 distinct positive integers, each of which is not divisible by divisor1. arr2 contains uniqueCnt2 distinct positive integers, each of which is not divisible by divisor2. No integer is present in both arr1 and arr2.

Given divisor1, divisor2, uniqueCnt1, and uniqueCnt2, return the minimum possible maximum integer that can be present in either array. Example 1: Input: divisor1 = 2, divisor2 = 7, uniqueCnt1 = 1, uniqueCnt2 = 3 Output: 4 Explanation: We can distribute the first 4 natural numbers into arr1 and arr2. arr1 = [1] and arr2 = [2,3,4]. We can see that both arrays satisfy all the conditions. Since the maximum value is 4, we return it.

Example 2: Input: divisor1 = 3, divisor2 = 5, uniqueCnt1 = 2, uniqueCnt2 = 1 Output: 3 Explanation: Here arr1 = [1,2], and arr2 = [3] satisfy all conditions.

Since the maximum value is 3, we return it. Example 3: Input: divisor1 = 2, divisor2 = 4, uniqueCnt1 = 8, uniqueCnt2 = 2 Output: 15 Explanation: Here, the final possible arrays can be arr1 = [1,3,5,7,9,11,13,15], and arr2 = [2,6]. It can be shown that it is not possible to obtain a lower maximum satisfying all conditions.

Constraints:

2 <= divisor1, divisor2 <= 105 1 <= uniqueCnt1, uniqueCnt2 < 109 2 <= uniqueCnt1 + uniqueCnt2 <= 109

==

Problem Number: 1552 URL: https://leetcode.com/problems/take-k-of-each-character-from-left-and-right Title: 2516. Take K of Each Character From Left and Right Problem Description: You are given a string s consisting of the characters 'a', 'b', and 'c' and a non-negative integer k. Each minute, you may take either the leftmost character of s, or the rightmost character of s. Return the minimum number of minutes needed for you to take at least k of each character, or return -1 if it is not possible to take k of each character. Example 1: Input: s = "aabaaaacaabc", k = 2 Output: 8 Explanation: Take three characters from the left of s. You now have two 'a' characters, and one 'b' character. Take five characters from the right of s. You now have four 'a' characters, two 'b' characters, and two 'c' characters. A total of 3 + 5 = 8 minutes is needed. It can be proven that 8 is the minimum number of minutes needed.

Example 2: Input: s = "a", k = 1 Output: -1 Explanation: It is not possible to take one 'b' or 'c' so return -1.

Constraints:

1 <= s.length <= 105 s consists of only the letters 'a', 'b', and 'c'. 0 <= k <= s.length

==

Problem Number: 1553 URL: https://leetcode.com/problems/maximum-tastiness-of-candy-basket Title: 2517. Maximum Tastiness of Candy Basket Problem Description: You are given an array of positive integers price where price[i] denotes the price of the ith candy and a positive integer k. The store sells baskets of k distinct candies. The tastiness of a candy basket is the smallest absolute difference of the prices of any two candies in the basket. Return the maximum tastiness of a candy basket. Example 1: Input: price = [13,5,1,8,21,2], k = 3 Output: 8 Explanation: Choose the candies with the prices [13,5,21]. The tastiness of the candy basket is: min(|13 - 5|, |13 - 21|, |5 - 21|) = min(8, 8, 16) = 8. It can be proven that 8 is the maximum tastiness that can be achieved.

Example 2: Input: price = [1,3,1], k = 2 Output: 2 Explanation: Choose the candies with the prices [1,3]. The tastiness of the candy basket is: min(|1 - 3|)

= min(2) = 2. It can be proven that 2 is the maximum tastiness that can be achieved.

Example 3: Input: price = [7,7,7,7], k = 2 Output: 0 Explanation: Choosing any two distinct candies from the candies we have will result in a tastiness of 0.

Constraints:

2 <= k <= price.length <= 105 1 <= price[i] <= 109

==
Problem Number: 1554 URL: https://leetcode.com/problems/distinct-prime-factors-of-product-of-array Title: 2521. Distinct Prime Factors of Product of Array Problem Description: Given an array of positive integers nums, return the number of distinct prime factors in the product of the elements of nums. Note that:

A number greater than 1 is called prime if it is divisible by only 1 and itself. An integer val1 is a factor of another integer val2 if val2 / val1 is an integer.

Example 1: Input: nums = [2,4,3,7,10,6] Output: 4 Explanation: The product of all the elements in nums is: 2 * 4 * 3 * 7 * 10 * 6 = 10080 = 25 * 32 * 5 * 7. There are 4 distinct prime factors so we return 4.

Example 2: Input: nums = [2,4,8,16] Output: 1 Explanation: The product of all the elements in nums is: 2 * 4 * 8 * 16 = 1024 = 210. There is 1 distinct prime factor so we return 1.

Constraints:

1 <= nums.length <= 104 2 <= nums[i] <= 1000

==
Problem Number: 1555 URL: https://leetcode.com/problems/partition-string-into-substrings-with-values-at-most-k Title: 2522. Partition String Into Substrings With Values at Most K Problem Description: You are given a string s consisting of digits from 1 to 9 and an integer k. A partition of a string s is called good if:

Each digit of s is part of exactly one substring. The value of each substring is less than or equal to k.

Return the minimum number of substrings in a good partition of s. If no good partition of s exists, return -1. Note that:

The value of a string is its result when interpreted as an integer. For example, the value of "123" is 123 and the value of "1" is 1. A substring is a contiguous sequence of characters within a string.

Example 1: Input: s = "165462", k = 60 Output: 4 Explanation: We can partition the string into substrings "16", "54", "6", and "2". Each substring has a value less than or equal to k = 60. It can be shown that we cannot partition the string into less than 4 substrings.

Example 2: Input: s = "238182", k = 5 Output: -1 Explanation: There is no good partition for this string.

Constraints:

1 <= s.length <= 105 s[i] is a digit from '1' to '9'. 1 <= k <= 109

==

Problem Number: 1556 URL: https://leetcode.com/problems/closest-prime-numbers-in-range Title: 2523. Closest Prime Numbers in Range Problem Description: Given two positive integers left and right, find the two integers num1 and num2 such that:

left <= nums1 < nums2 <= right . nums1 and nums2 are both prime numbers. nums2 - nums1 is the minimum amongst all other pairs satisfying the above conditions.

Return the positive integer array ans = [nums1, nums2]. If there are multiple pairs satisfying these conditions, return the one with the minimum nums1 value or [-1, -1] if such numbers do not exist. A number greater than 1 is called prime if it is only divisible by 1 and itself. Example 1: Input: left = 10, right = 19 Output: [11,13] Explanation: The prime numbers between 10 and 19 are 11, 13, 17, and 19. The closest gap between any pair is 2, which can be achieved by [11,13] or [17,19]. Since 11 is smaller than 17, we return the first pair.

Example 2: Input: left = 4, right = 6 Output: [-1,-1] Explanation: There exists only one prime number in the given range, so the conditions cannot be satisfied.

Constraints:

1 <= left <= right <= 106

==

Problem Number: 1557 URL: https://leetcode.com/problems/find-consecutive-integers-from-a-data-stream Title: 2526. Find Consecutive Integers from a Data Stream Problem Description: For a stream of integers, implement a data structure that checks if the last k integers parsed in the stream are equal to value. Implement the DataStream class:

DataStream(int value, int k) Initializes the object with an empty integer stream and the two integers value and k. boolean consec(int num) Adds num to the stream of integers. Returns true if the last k integers are equal to value, and false otherwise. If there are less than k integers, the condition does not hold true, so returns false.

Example 1: Input ["DataStream", "consec", "consec", "consec", "consec"] [[4, 3], [4], [4], [4], [3]] Output [null, false, false, true, false]

Explanation DataStream dataStream = new DataStream(4, 3); //value = 4, k = 3 dataStream.consec(4); // Only 1 integer is parsed, so returns False. dataStream.consec(4); // Only 2 integers are parsed. // Since 2 is less than k, returns False. dataStream.consec(4); // The 3 integers parsed are all equal to value, so returns True. dataStream.consec(3); // The last k integers parsed in the stream are [4,4,3]. // Since 3 is not equal to value, it returns False.

Constraints:

1 <= value, num <= 109 1 <= k <= 105 At most 105 calls will be made to consec.

==
Problem Number: 1558 URL: https://leetcode.com/problems/find-xor-beauty-of-array Title: 2527. Find Xor-Beauty of Array Problem Description: You are given a 0-indexed integer array nums. The effective value of three indices i, j, and k is defined as ((nums[i] | nums[j]) & nums[k]). The xor-beauty of the array is the XORing of the effective values of all the possible triplets of indices (i, j, k) where 0 <= i, j, k < n. Return the xor-beauty of nums. Note that:

val1 | val2 is bitwise OR of val1 and val2. val1 & val2 is bitwise AND of val1 and val2.

Example 1: Input: nums = [1,4] Output: 5 Explanation: The triplets and their corresponding effective values are listed below: - (0,0,0) with effective value ((1 | 1) & 1) = 1 - (0,0,1) with effective value ((1 | 1) & 4) = 0 - (0,1,0) with effective value ((1 | 4) & 1) = 1 - (0,1,1) with effective value ((1 | 4) & 4) = 4 - (1,0,0) with effective value ((4 | 1) & 1) = 1 - (1,0,1) with effective value ((4 | 1) & 4) = 4 - (1,1,0) with effective value ((4 | 4) & 1) = 0 - (1,1,1) with effective value ((4 | 4) & 4) = 4 Xor-beauty of array will be bitwise XOR of all beauties = 1 ^ 0 ^ 1 ^ 4 ^ 1 ^ 4 ^ 0 ^ 4 = 5. Example 2: Input: nums = [15,45,20,2,34,35,5,44,32,30] Output: 34 Explanation: The xor-beauty of the given array is 34.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1559 URL: https://leetcode.com/problems/maximal-score-after-applying-k-operations Title: 2530. Maximal Score After Applying K Operations Problem Description: You are given a 0-indexed integer array nums and an integer k. You have a starting score of 0. In one operation:

choose an index i such that 0 <= i < nums.length, increase your score by nums[i], and replace nums[i] with ceil(nums[i] / 3).

Return the maximum possible score you can attain after applying exactly k operations. The ceiling function ceil(val) is the least integer greater than or equal to val. Example 1: Input: nums = [10,10,10,10,10], k = 5 Output: 50 Explanation: Apply the operation to each array element exactly once. The final score is 10 + 10 + 10 + 10 + 10 = 50.

Example 2: Input: nums = [1,10,3,3,3], k = 3 Output: 17 Explanation: You can do the following operations: Operation 1: Select i = 1, so nums becomes [1,4,3,3,3]. Your score increases by 10. Operation 2: Select i = 1, so nums becomes [1,2,3,3,3]. Your score increases by 4. Operation 3: Select i = 2, so nums becomes [1,1,1,3,3]. Your score increases by 3. The final score is 10 + 4 + 3 = 17.

Constraints:

1 <= nums.length, k <= 105 1 <= nums[i] <= 109

==
Problem Number: 1560 URL: https://leetcode.com/problems/make-number-of-distinct-characters-equal Title: 2531. Make Number of Distinct Characters Equal Problem Description: You are given two 0-indexed strings word1 and word2. A move consists of choosing two indices i and j such that 0 <= i < word1.length and 0 <= j < word2.length and swapping word1[i] with word2[j]. Return true if it is possible to get the number of distinct characters in word1 and word2 to be equal with exactly one move. Return false otherwise. Example 1: Input: word1 = "ac", word2 = "b" Output: false Explanation: Any pair of swaps would yield two distinct characters in the first string, and one in the second string.

Example 2: Input: word1 = "abcc", word2 = "aab" Output: true Explanation: We swap index 2 of the first string with index 0 of the second string. The resulting strings are word1 = "abac" and word2 = "cab", which both have 3 distinct characters.

Example 3: Input: word1 = "abcde", word2 = "fghij" Output: true Explanation: Both resulting strings will have 5 distinct characters, regardless of which indices we swap.

Constraints:

1 <= word1.length, word2.length <= 105 word1 and word2 consist of only lowercase English letters.

==
Problem Number: 1561 URL: https://leetcode.com/problems/increment-submatrices-by-one Title: 2536. Increment Submatrices by One Problem Description: You are given a positive integer n, indicating that we initially have an n x n 0-indexed integer matrix mat filled with zeroes. You are also given a 2D integer array query. For each query[i] = [row1i, col1i, row2i, col2i], you should do the following operation:

Add 1 to every element in the submatrix with the top left corner (row1i, col1i) and the bottom right corner (row2i, col2i). That is, add 1 to mat[x][y] for all row1i <= x <= row2i and col1i <= y <= col2i.

Return the matrix mat after performing every query. Example 1:

Input: n = 3, queries = [[1,1,2,2],[0,0,1,1]] Output: [[1,1,0],[1,2,1],[0,1,1]] Explanation: The diagram above shows the initial matrix, the matrix after the first query, and the matrix after the second query. - In the first query, we add 1 to every element in the submatrix with the top left corner (1, 1) and bottom right corner (2, 2). - In the second query, we add 1 to every element in the submatrix with the top left corner (0, 0) and bottom right corner (1, 1).

Example 2:

Input: n = 2, queries = [[0,0,1,1]] Output: [[1,1],[1,1]] Explanation: The diagram above shows the initial matrix and the matrix after the first query. - In the first query we add 1 to every element in the matrix.

Constraints:

1 <= n <= 500 1 <= queries.length <= 104 0 <= row1i <= row2i < n 0 <= col1i <= col2i < n

==
Problem Number: 1562 URL: https://leetcode.com/problems/count-the-number-of-good-subarrays Title: 2537. Count the Number of Good Subarrays Problem Description: Given an integer array nums and an integer k, return the number of good subarrays of nums. A subarray arr is good if it there are at least k pairs of indices (i, j) such that i < j and arr[i] == arr[j]. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,1,1,1,1], k = 10 Output: 1 Explanation: The only good subarray is the array nums itself.

Example 2: Input: nums = [3,1,4,3,2,2,4], k = 2 Output: 4 Explanation: There are 4 different good subarrays: - [3,1,4,3,2,2] that has 2 pairs. - [3,1,4,3,2,2,4] that has 3 pairs. - [1,4,3,2,2,4] that has 2 pairs. - [4,3,2,2,4] that has 2 pairs.

Constraints:

1 <= nums.length <= 105 1 <= nums[i], k <= 109

==
Problem Number: 1563 URL: https://leetcode.com/problems/minimum-operations-to-make-array-equal-ii Title: 2541. Minimum Operations to Make Array Equal II Problem Description: You are given two integer arrays nums1 and nums2 of equal length n and an integer k. You can perform the following operation on nums1:

Choose two indexes i and j and increment nums1[i] by k and decrement nums1[j] by k. In other words, nums1[i] = nums1[i] + k and nums1[j] = nums1[j] - k.

nums1 is said to be equal to nums2 if for all indices i such that 0 <= i < n, nums1[i] == nums2[i]. Return the minimum number of operations required to make nums1 equal to nums2. If it is impossible to make them equal, return -1. Example 1: Input: nums1 = [4,3,1,4], nums2 = [1,3,7,1], k = 3 Output: 2 Explanation: In 2 operations, we can transform nums1 to nums2. 1st operation:

i = 2, j = 0. After applying the operation, nums1 = [1,3,4,4]. 2nd operation: i = 2, j = 3. After applying the operation, nums1 = [1,3,7,1]. One can prove that it is impossible to make arrays equal in fewer operations. Example 2: Input: nums1 = [3,8,5,2], nums2 = [2,4,1,6], k = 1 Output: -1 Explanation: It can be proved that it is impossible to make the two arrays equal.

Constraints:

n == nums1.length == nums2.length 2 <= n <= 105 0 <= nums1[i], nums2[j] <= 109 0 <= k <= 105

==
Problem Number: 1564 URL: https://leetcode.com/problems/maximum-subsequence-score Title: 2542. Maximum Subsequence Score Problem Description: You are given two 0-indexed integer arrays nums1 and nums2 of equal length n and a positive integer k. You must choose a subsequence of indices from nums1 of length k. For chosen indices i0, i1, ..., ik - 1, your score is defined as:

The sum of the selected elements from nums1 multiplied with the minimum of the selected elements from nums2. It can defined simply as: (nums1[i0] + nums1[i1] +...+ nums1[ik - 1]) * min(nums2[i0] , nums2[i1], ... ,nums2[ik - 1]).

Return the maximum possible score. A subsequence of indices of an array is a set that can be derived from the set {0, 1, ..., n-1} by deleting some or no elements. Example 1: Input: nums1 = [1,3,3,2], nums2 = [2,1,3,4], k = 3 Output: 12 Explanation: The four possible subsequence scores are: - We choose the indices 0, 1, and 2 with score = (1+3+3) * min(2,1,3) = 7. - We choose the indices 0, 1, and 3 with score = (1+3+2) * min(2,1,4) = 6. - We choose the indices 0, 2, and 3 with score = (1+3+2) * min(2,3,4) = 12. - We choose the indices 1, 2, and 3 with score = (3+3+2) * min(1,3,4) = 8. Therefore, we return the max score, which is 12.

Example 2: Input: nums1 = [4,2,3,1,1], nums2 = [7,5,10,9,6], k = 1 Output: 30 Explanation: Choosing index 2 is optimal: nums1[2] * nums2[2] = 3 * 10 = 30 is the maximum possible score.

Constraints:

n == nums1.length == nums2.length 1 <= n <= 105 0 <= nums1[i], nums2[j] <= 105 1 <= k <= n

==
Problem Number: 1565 URL: https://leetcode.com/problems/sort-the-students-by-their-kth-score Title: 2545. Sort the Students by Their Kth Score Problem Description: There is a class with m students and n exams. You are given a 0-indexed m x n integer matrix score, where each row represents one student and score[i][j] denotes the score the ith student got in the jth exam. The matrix score contains distinct integers only. You are also given an integer k. Sort the students (i.e., the rows of the matrix) by their scores in the

kth (0-indexed) exam from the highest to the lowest. Return the matrix after sorting it. Example 1:

Input: score = [[10,6,9,1],[7,5,11,2],[4,8,3,15]], k = 2 Output: [[7,5,11,2],[10,6,9,1],[4,8,3,15]] Explanation: In the above diagram, S denotes the student, while E denotes the exam. - The student with index 1 scored 11 in exam 2, which is the highest score, so they got first place. - The student with index 0 scored 9 in exam 2, which is the second highest score, so they got second place. - The student with index 2 scored 3 in exam 2, which is the lowest score, so they got third place.

Example 2:

Input: score = [[3,4],[5,6]], k = 0 Output: [[5,6],[3,4]] Explanation: In the above diagram, S denotes the student, while E denotes the exam. - The student with index 1 scored 5 in exam 0, which is the highest score, so they got first place. - The student with index 0 scored 3 in exam 0, which is the lowest score, so they got second place.

Constraints:

m == score.length n == score[i].length 1 <= m, n <= 250 1 <= score[i][j] <= 105 score consists of distinct integers. 0 <= k < n

==
Problem Number: 1566 URL: https://leetcode.com/problems/apply-bitwise-operations-to-make-strings-equal Title: 2546. Apply Bitwise Operations to Make Strings Equal Problem Description: You are given two 0-indexed binary strings s and target of the same length n. You can do the following operation on s any number of times:

Choose two different indices i and j where 0 <= i, j < n. Simultaneously, replace s[i] with (s[i] OR s[j]) and s[j] with (s[i] XOR s[j]).

For example, if s = "0110", you can choose i = 0 and j = 2, then simultaneously replace s[0] with (s[0] OR s[2] = 0 OR 1 = 1), and s[2] with (s[0] XOR s[2] = 0 XOR 1 = 1), so we will have s = "1110". Return true if you can make the string s equal to target, or false otherwise. Example 1: Input: s = "1010", target = "0110" Output: true Explanation: We can do the following operations: - Choose i = 2 and j = 0. We have now s = "0010". - Choose i = 2 and j = 1. We have now s = "0110". Since we can make s equal to target, we return true.

Example 2: Input: s = "11", target = "00" Output: false Explanation: It is not possible to make s equal to target with any number of operations.

Constraints:

n == s.length == target.length 2 <= n <= 105 s and target consist of only the digits 0 and 1.

==
Problem Number: 1567 URL: https://leetcode.com/problems/count-collisions-of-monkeys-on-a-polygon Title: 2550. Count Collisions of Monkeys on a

Polygon Problem Description: There is a regular convex polygon with n vertices. The vertices are labeled from 0 to n - 1 in a clockwise direction, and each vertex has exactly one monkey. The following figure shows a convex polygon of 6 vertices.

Each monkey moves simultaneously to a neighboring vertex. A neighboring vertex for a vertex i can be:

the vertex (i + 1) % n in the clockwise direction, or the vertex (i - 1 + n) % n in the counter-clockwise direction.

A collision happens if at least two monkeys reside on the same vertex after the movement or intersect on an edge. Return the number of ways the monkeys can move so that at least one collision happens. Since the answer may be very large, return it modulo 109 + 7. Note that each monkey can only move once. Example 1: Input: n = 3 Output: 6 Explanation: There are 8 total possible movements. Two ways such that they collide at some point are: - Monkey 1 moves in a clockwise direction; monkey 2 moves in an anticlockwise direction; monkey 3 moves in a clockwise direction. Monkeys 1 and 2 collide. - Monkey 1 moves in an anticlockwise direction; monkey 2 moves in an anticlockwise direction; monkey 3 moves in a clockwise direction. Monkeys 1 and 3 collide. It can be shown 6 total movements result in a collision.

Example 2: Input: n = 4 Output: 14 Explanation: It can be shown that there are 14 ways for the monkeys to collide.

Constraints:

3 <= n <= 109

==

Problem Number: 1568 URL: https://leetcode.com/problems/maximum-number-of-integers-to-choose-from-a-range-i Title: 2554. Maximum Number of Integers to Choose From a Range I Problem Description: You are given an integer array banned and two integers n and maxSum. You are choosing some number of integers following the below rules:

The chosen integers have to be in the range [1, n]. Each integer can be chosen at most once. The chosen integers should not be in the array banned. The sum of the chosen integers should not exceed maxSum.

Return the maximum number of integers you can choose following the mentioned rules. Example 1: Input: banned = [1,6,5], n = 5, maxSum = 6 Output: 2 Explanation: You can choose the integers 2 and 4. 2 and 4 are from the range [1, 5], both did not appear in banned, and their sum is 6, which did not exceed maxSum.

Example 2: Input: banned = [1,2,3,4,5,6,7], n = 8, maxSum = 1 Output: 0 Explanation: You cannot choose any integer while following the mentioned conditions.

Example 3: Input: banned = [11], n = 7, maxSum = 50 Output: 7 Explanation: You can choose the integers 1, 2, 3, 4, 5, 6, and 7. They are from the range [1, 7], all did not appear in banned, and their sum is 28, which did not exceed maxSum.

Constraints:

1 <= banned.length <= 104 1 <= banned[i], n <= 104 1 <= maxSum <= 109

== Problem Number: 1569 URL: https://leetcode.com/problems/maximize-win-from-two-segments Title: 2555. Maximize Win From Two Segments Problem Description: There are some prizes on the X-axis. You are given an integer array prizePositions that is sorted in non-decreasing order, where prizePositions[i] is the position of the ith prize. There could be different prizes at the same position on the line. You are also given an integer k. You are allowed to select two segments with integer endpoints. The length of each segment must be k. You will collect all prizes whose position falls within at least one of the two selected segments (including the endpoints of the segments). The two selected segments may intersect.

For example if k = 2, you can choose segments [1, 3] and [2, 4], and you will win any prize i that satisfies 1 <= prizePositions[i] <= 3 or 2 <= prizePositions[i] <= 4.

Return the maximum number of prizes you can win if you choose the two segments optimally. Example 1: Input: prizePositions = [1,1,2,2,3,3,5], k = 2 Output: 7 Explanation: In this example, you can win all 7 prizes by selecting two segments [1, 3] and [3, 5].

Example 2: Input: prizePositions = [1,2,3,4], k = 0 Output: 2 Explanation: For this example, one choice for the segments is [3, 3] and [4, 4], and you will be able to get 2 prizes.

Constraints:

1 <= prizePositions.length <= 105 1 <= prizePositions[i] <= 109 0 <= k <= 109 prizePositions is sorted in non-decreasing order.

== Problem Number: 1570 URL: https://leetcode.com/problems/disconnect-path-in-a-binary-matrix-by-at-most-one-flip Title: 2556. Disconnect Path in a Binary Matrix by at Most One Flip Problem Description: You are given a 0-indexed m x n binary matrix grid. You can move from a cell (row, col) to any of the cells (row + 1, col) or (row, col + 1) that has the value 1. The matrix is disconnected if there is no path from (0, 0) to (m - 1, n - 1). You can flip the value of at most one (possibly none) cell. You cannot flip the cells (0, 0) and (m - 1, n - 1). Return true if it is possible to make the matrix disconnect or false otherwise. Note that flipping a cell changes its value from 0 to 1 or from 1 to 0. Example 1:

Input: grid = [[1,1,1],[1,0,0],[1,1,1]] Output: true Explanation: We can change the cell shown in the diagram above. There is no path from (0, 0) to (2, 2) in the resulting grid.

Example 2:

Input: grid = [[1,1,1],[1,0,1],[1,1,1]] Output: false Explanation: It is not possible to change at most one cell such that there is not path from (0, 0) to (2, 2).

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 1000 1 <= m * n <= 105 grid[i][j] is either 0 or 1. grid[0][0] == grid[m - 1][n - 1] == 1

==
Problem Number: 1571 URL: https://leetcode.com/problems/count-vowel-strings-in-ranges Title: 2559. Count Vowel Strings in Ranges Problem Description: You are given a 0-indexed array of strings words and a 2D array of integers queries. Each query queries[i] = [li, ri] asks us to find the number of strings present in the range li to ri (both inclusive) of words that start and end with a vowel. Return an array ans of size queries.length, where ans[i] is the answer to the ith query. Note that the vowel letters are 'a', 'e', 'i', 'o', and 'u'. Example 1: Input: words = ["aba","bcb","ece","aa","e"], queries = [[0,2],[1,4],[1,1]] Output: [2,3,0] Explanation: The strings starting and ending with a vowel are "aba", "ece", "aa" and "e". The answer to the query [0,2] is 2 (strings "aba" and "ece"). to query [1,4] is 3 (strings "ece", "aa", "e"). to query [1,1] is 0. We return [2,3,0].

Example 2: Input: words = ["a","e","i"], queries = [[0,2],[0,1],[2,2]] Output: [3,2,1] Explanation: Every string satisfies the conditions, so we return [3,2,1]. Constraints:

1 <= words.length <= 105 1 <= words[i].length <= 40 words[i] consists only of lowercase English letters. sum(words[i].length) <= 3 * 105 1 <= queries.length <= 105 0 <= li <= ri < words.length

==
Problem Number: 1572 URL: https://leetcode.com/problems/house-robber-iv Title: 2560. House Robber IV Problem Description: There are several consecutive houses along a street, each of which has some money inside. There is also a robber, who wants to steal money from the homes, but he refuses to steal from adjacent homes. The capability of the robber is the maximum amount of money he steals from one house of all the houses he robbed. You are given an integer array nums representing how much money is stashed in each house. More formally, the ith house from the left has nums[i] dollars. You are also given an integer k, representing the minimum number of houses the robber will steal from. It is always possible to steal at least k houses. Return the

minimum capability of the robber out of all the possible ways to steal at least k houses. Example 1: Input: nums = [2,3,5,9], k = 2 Output: 5 Explanation: There are three ways to rob at least 2 houses: - Rob the houses at indices 0 and 2. Capability is max(nums[0], nums[2]) = 5. - Rob the houses at indices 0 and 3. Capability is max(nums[0], nums[3]) = 9. - Rob the houses at indices 1 and 3. Capability is max(nums[1], nums[3]) = 9. Therefore, we return min(5, 9, 9) = 5.

Example 2: Input: nums = [2,7,9,3,1], k = 2 Output: 2 Explanation: There are 7 ways to rob the houses. The way which leads to minimum capability is to rob the house at index 0 and 4. Return max(nums[0], nums[4]) = 2.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= k <= (nums.length + 1)/2

==
Problem Number: 1573 URL: https://leetcode.com/problems/count-the-number-of-fair-pairs Title: 2563. Count the Number of Fair Pairs Problem Description: Given a 0-indexed integer array nums of size n and two integers lower and upper, return the number of fair pairs. A pair (i, j) is fair if:

0 <= i < j < n, and lower <= nums[i] + nums[j] <= upper

Example 1: Input: nums = [0,1,7,4,4,5], lower = 3, upper = 6 Output: 6 Explanation: There are 6 fair pairs: (0,3), (0,4), (0,5), (1,3), (1,4), and (1,5).

Example 2: Input: nums = [1,7,9,2,5], lower = 11, upper = 11 Output: 1 Explanation: There is a single fair pair: (2,3).

Constraints:

1 <= nums.length <= 105 nums.length == n -109 <= nums[i] <= 109 -109 <= lower <= upper <= 109

==
Problem Number: 1574 URL: https://leetcode.com/problems/substring-xor-queries Title: 2564. Substring XOR Queries Problem Description: You are given a binary string s, and a 2D integer array queries where queries[i] = [firsti, secondi]. For the ith query, find the shortest substring of s whose decimal value, val, yields secondi when bitwise XORed with firsti. In other words, val ^ firsti == secondi. The answer to the ith query is the endpoints (0-indexed) of the substring [lefti, righti] or [-1, -1] if no such substring exists. If there are multiple answers, choose the one with the minimum lefti. Return an array ans where ans[i] = [lefti, righti] is the answer to the ith query. A substring is a contiguous non-empty sequence of characters within a string. Example 1: Input: s = "101101", queries = [[0,5],[1,2]] Output: [[0,2],[2,3]] Explanation: For the first query the substring in range [0,2] is "101" which has a decimal value of 5, and 5 ^ 0 = 5, hence the answer to the first query is [0,2]. In the

second query, the substring in range [2,3] is "11", and has a decimal value of 3, and 3 ^ 1 = 2. So, [2,3] is returned for the second query.

Example 2: Input: s = "0101", queries = [[12,8]] Output: [[-1,-1]] Explanation: In this example there is no substring that answers the query, hence [-1,-1] is returned.

Example 3: Input: s = "1", queries = [[4,5]] Output: [[0,0]] Explanation: For this example, the substring in range [0,0] has a decimal value of 1, and 1 ^ 4 = 5. So, the answer is [0,0].

Constraints:

1 <= s.length <= 104 s[i] is either '0' or '1'. 1 <= queries.length <= 105 0 <= firsti, secondi <= 109

==

Problem Number: 1575 URL: https://leetcode.com/problems/minimum-score-by-changing-two-elements Title: 2567. Minimum Score by Changing Two Elements Problem Description: You are given a 0-indexed integer array nums.

The low score of nums is the minimum value of |nums[i] - nums[j]| over all 0 <= i < j < nums.length. The high score of nums is the maximum value of |nums[i] - nums[j]| over all 0 <= i < j < nums.length. The score of nums is the sum of the high and low scores of nums.

To minimize the score of nums, we can change the value of at most two elements of nums. Return the minimum possible score after changing the value of at most two elements of nums. Note that |x| denotes the absolute value of x. Example 1: Input: nums = [1,4,3] Output: 0 Explanation: Change value of nums[1] and nums[2] to 1 so that nums becomes [1,1,1]. Now, the value of |nums[i] - nums[j]| is always equal to 0, so we return 0 + 0 = 0.

Example 2: Input: nums = [1,4,7,8,5] Output: 3 Explanation: Change nums[0] and nums[1] to be 6. Now nums becomes [6,6,7,8,5]. Our low score is achieved when i = 0 and j = 1, in which case |nums[i] - nums[j]| = |6 - 6| = 0. Our high score is achieved when i = 3 and j = 4, in which case |nums[i] - nums[j]| = |8 - 5| = 3. The sum of our high and low score is 3, which we can prove to be minimal.

Constraints:

3 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 1576 URL: https://leetcode.com/problems/minimum-impossible-or Title: 2568. Minimum Impossible OR Problem Description: You are given a 0-indexed integer array nums. We say that an integer x is expressible from nums if there exist some integers 0 <= index1 < index2 < ... < indexk < nums.length for which nums[index1] | nums[index2] | ... | nums[indexk] = x. In other words, an integer is expressible if it can be written as the bitwise OR of some subsequence of nums. Return the minimum positive non-zero integer that is not expressible from nums.   Example 1: Input: nums = [2,1] Output: 4 Explanation: 1 and 2 are already present in the array. We know that 3 is expressible, since nums[0] | nums[1] = 2 | 1 = 3. Since 4 is not expressible, we return 4.

Example 2: Input: nums = [5,3,2] Output: 1 Explanation: We can show that 1 is the smallest number that is not expressible.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 1577 URL: https://leetcode.com/problems/minimum-operations-to-reduce-an-integer-to-0 Title: 2571. Minimum Operations to Reduce an Integer to 0 Problem Description: You are given a positive integer n, you can do the following operation any number of times:

Add or subtract a power of 2 from n.

Return the minimum number of operations to make n equal to 0. A number x is power of 2 if x == 2i where i >= 0.   Example 1: Input: n = 39 Output: 3 Explanation: We can do the following operations: - Add 20 = 1 to n, so now n = 40. - Subtract 23 = 8 from n, so now n = 32. - Subtract 25 = 32 from n, so now n = 0. It can be shown that 3 is the minimum number of operations we need to make n equal to 0.

Example 2: Input: n = 54 Output: 3 Explanation: We can do the following operations: - Add 21 = 2 to n, so now n = 56. - Add 23 = 8 to n, so now n = 64. - Subtract 26 = 64 from n, so now n = 0. So the minimum number of operations is 3.

Constraints:

1 <= n <= 105

==

Problem Number: 1578 URL: https://leetcode.com/problems/count-the-number-of-square-free-subsets Title: 2572. Count the Number of Square-Free Subsets Problem Description: You are given a positive integer 0-indexed array nums. A subset of the array nums is square-free if the product of its elements is a square-free integer. A square-free integer is an integer that is divisible by no square number other than 1. Return the number of square-free non-empty subsets of the array nums. Since the answer may be too large, return it modulo 109 + 7. A non-empty subset of nums is an array that can be obtained by deleting some (possibly none but not all) elements from nums. Two subsets are different if and only if the chosen indices to delete are different.   Example 1: Input: nums = [3,4,4,5] Output: 3 Explanation: There are 3 square-free subsets in this example: - The subset consisting of the 0th element [3]. The product of its elements is 3, which is a square-free integer. - The subset consisting of the 3rd element [5]. The product of its elements is 5, which is a square-free integer. - The subset consisting of 0th and 3rd elements [3,5]. The product of its elements is 15, which is a square-free integer. It can be proven that there are no more than 3 square-free subsets in the given array. Example 2: Input: nums = [1] Output: 1 Explanation: There is 1 square-free subset in this example: - The subset consisting of the 0th element [1]. The product of its elements is 1, which is a square-free integer. It can be proven that there is no more than 1 square-free subset in the given array.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 30

==
Problem Number: 1579 URL: https://leetcode.com/problems/find-the-divisibility-array-of-a-string Title: 2575. Find the Divisibility Array of a String Problem Description: You are given a 0-indexed string word of length n consisting of digits, and a positive integer m. The divisibility array div of word is an integer array of length n such that:

div[i] = 1 if the numeric value of word[0,...,i] is divisible by m, or div[i] = 0 otherwise.

Return the divisibility array of word. Example 1: Input: word = "998244353", m = 3 Output: [1,1,0,0,0,1,1,0,0] Explanation: There are only 4 prefixes that are divisible by 3: "9", "99", "998244", and "9982443".

Example 2: Input: word = "1010", m = 10 Output: [0,1,0,1] Explanation: There are only 2 prefixes that are divisible by 10: "10", and "1010".

Constraints:

1 <= n <= 105 word.length == n word consists of digits from 0 to 9 1 <= m <= 109

==
Problem Number: 1580 URL: https://leetcode.com/problems/find-the-maximum-number-of-marked-indices Title: 2576. Find the Maximum Number of Marked Indices Problem Description: You are given a 0-indexed integer array nums. Initially, all of the indices are unmarked. You are allowed to make this operation any number of times:

Pick two different unmarked indices i and j such that 2 * nums[i] <= nums[j], then mark i and j.

Return the maximum possible number of marked indices in nums using the above operation any number of times. Example 1: Input: nums = [3,5,2,4] Output: 2 Explanation: In the first operation: pick i = 2 and j = 1, the operation is allowed because 2 * nums[2] <= nums[1]. Then mark index 2 and 1. It can be shown that there's no other valid operation so the answer is 2.

Example 2: Input: nums = [9,2,5,4] Output: 4 Explanation: In the first operation: pick i = 3 and j = 0, the operation is allowed because 2 * nums[3] <= nums[0]. Then mark index 3 and 0. In the second operation: pick i = 1 and j = 2, the operation is allowed because 2 * nums[1] <= nums[2]. Then mark index 1 and 2. Since there is no other operation, the answer is 4.

Example 3: Input: nums = [7,6,8] Output: 0 Explanation: There is no valid operation to do, so the answer is 0.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 1581 URL: https://leetcode.com/problems/count-total-number-of-colored-cells Title: 2579. Count Total Number of Colored Cells Problem Description: There exists an infinitely large two-dimensional grid of uncolored unit cells. You are given a positive integer n, indicating that you must do the following routine for n minutes:

At the first minute, color any arbitrary unit cell blue. Every minute thereafter, color blue every uncolored cell that touches a blue cell.

Below is a pictorial representation of the state of the grid after minutes 1, 2, and 3.

Return the number of colored cells at the end of n minutes. Example 1: Input: n = 1 Output: 1 Explanation: After 1 minute, there is only 1 blue cell, so we return 1.

Example 2: Input: n = 2 Output: 5 Explanation: After 2 minutes, there are 4 colored cells on the boundary and 1 in the center, so we return 5.

Constraints:

1 <= n <= 105

==

Problem Number: 1582 URL: https://leetcode.com/problems/count-ways-to-group-overlapping-ranges Title: 2580. Count Ways to Group Overlapping Ranges Problem Description: You are given a 2D integer array ranges where ranges[i] = [starti, endi] denotes that all integers between starti and endi (both inclusive) are contained in the ith range. You are to split ranges into two (possibly empty) groups such that:

Each range belongs to exactly one group. Any two overlapping ranges must belong to the same group.

Two ranges are said to be overlapping if there exists at least one integer that is present in both ranges.

For example, [1, 3] and [2, 5] are overlapping because 2 and 3 occur in both ranges.

Return the total number of ways to split ranges into two groups. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: ranges = [[6,10],[5,15]] Output: 2 Explanation: The two ranges are overlapping, so they must be in the same group. Thus, there are two possible ways: - Put both the ranges together in group 1. - Put both the ranges together in group 2.

Example 2: Input: ranges = [[1,3],[10,20],[2,5],[4,8]] Output: 4 Explanation: Ranges [1,3], and [2,5] are overlapping. So, they must be in the same group. Again, ranges [2,5] and [4,8] are also overlapping. So, they must also be in the same group. Thus, there are four possible ways to group them: - All the ranges in group 1. - All the ranges in group 2. - Ranges [1,3], [2,5], and [4,8] in group 1 and [10,20] in group 2. - Ranges [1,3], [2,5], and [4,8] in group 2 and [10,20] in group 1.

Constraints:

1 <= ranges.length <= 105 ranges[i].length == 2 0 <= starti <= endi <= 109

==
Problem Number: 1583 URL: https://leetcode.com/problems/kth-largest-sum-in-a-binary-tree Title: 2583. Kth Largest Sum in a Binary Tree Problem Description: You are given the root of a binary tree and a positive integer k. The level sum in the tree is the sum of the values of the nodes that are on the same level. Return the kth largest level sum in the tree (not necessarily distinct). If there are fewer than k levels in the tree, return -1. Note that two nodes are on the same level if they have the same distance from the root. Example 1:

Input: root = [5,8,9,2,1,3,7,4,6], k = 2 Output: 13 Explanation: The level sums are the following: - Level 1: 5. - Level 2: 8 + 9 = 17. - Level 3: 2 + 1 + 3 + 7 = 13. - Level 4: 4 + 6 = 10. The 2nd largest level sum is 13.

Example 2:

Input: root = [1,2,null,3], k = 1 Output: 3 Explanation: The largest level sum is 3.

Constraints:

The number of nodes in the tree is n. 2 <= n <= 105 1 <= Node.val <= 106 1 <= k <= n

==
Problem Number: 1584 URL: https://leetcode.com/problems/rearrange-array-to-maximize-prefix-score Title: 2587. Rearrange Array to Maximize Prefix Score Problem Description: You are given a 0-indexed integer array nums. You can rearrange the elements of nums to any order (including the given order). Let prefix be the array containing the prefix sums of nums after rearranging it.

In other words, prefix[i] is the sum of the elements from 0 to i in nums after rearranging it. The score of nums is the number of positive integers in the array prefix. Return the maximum score you can achieve. Example 1: Input: nums = [2,-1,0,1,-3,3,-3] Output: 6 Explanation: We can rearrange the array into nums = [2,3,1,-1,-3,0,-3]. prefix = [2,5,6,5,2,2,-1], so the score is 6. It can be shown that 6 is the maximum score we can obtain.

Example 2: Input: nums = [-2,-3,0] Output: 0 Explanation: Any rearrangement of the array will result in a score of 0.

Constraints:

1 <= nums.length <= 105 -106 <= nums[i] <= 106

==
Problem Number: 1585 URL: https://leetcode.com/problems/count-the-number-of-beautiful-subarrays Title: 2588. Count the Number of Beautiful Subarrays Problem Description: You are given a 0-indexed integer array nums. In one operation, you can:

Choose two different indices i and j such that 0 <= i, j < nums.length. Choose a non-negative integer k such that the kth bit (0-indexed) in the binary representation of nums[i] and nums[j] is 1. Subtract 2k from nums[i] and nums[j].

A subarray is beautiful if it is possible to make all of its elements equal to 0 after applying the above operation any number of times. Return the number of beautiful subarrays in the array nums. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [4,3,1,2,4] Output: 2 Explanation: There are 2 beautiful subarrays in nums: [4,3,1,2,4] and [4,3,1,2,4]. - We can make all elements in the subarray [3,1,2] equal to 0 in the following way: - Choose [3, 1, 2] and k = 1. Subtract 21 from both numbers. The subarray becomes [1, 1, 0]. - Choose [1, 1, 0] and k = 0. Subtract 20 from both numbers. The subarray becomes [0, 0, 0]. - We can make all elements in the subarray [4,3,1,2,4] equal to 0 in the following way: - Choose [4, 3, 1, 2, 4] and k = 2. Subtract 22 from both numbers. The subarray becomes [0, 3, 1, 2, 0]. - Choose [0, 3, 1, 2, 0] and k = 0. Subtract 20 from both numbers. The subarray becomes [0, 2, 0, 2, 0]. - Choose [0, 2, 0, 2, 0] and k = 1. Subtract 21 from both numbers. The subarray becomes [0, 0, 0, 0, 0].

Example 2: Input: nums = [1,10,4] Output: 0 Explanation: There are no beautiful subarrays in nums.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 106

==
Problem Number: 1586 URL: https://leetcode.com/problems/maximize-greatness-of-an-array Title: 2592. Maximize Greatness of an Array Problem Description: You are given a 0-indexed integer array nums. You are allowed to permute nums into a new array perm of your choosing. We define the

greatness of nums be the number of indices 0 <= i < nums.length for which perm[i] > nums[i]. Return the maximum possible greatness you can achieve after permuting nums.   Example 1: Input: nums = [1,3,5,2,1,3,1] Output: 4 Explanation: One of the optimal rearrangements is perm = [2,5,1,3,3,1,1]. At indices = 0, 1, 3, and 4, perm[i] > nums[i]. Hence, we return 4. Example 2: Input: nums = [1,2,3,4] Output: 3 Explanation: We can prove the optimal perm is [2,3,4,1]. At indices = 0, 1, and 2, perm[i] > nums[i]. Hence, we return 3.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==
Problem Number: 1587 URL: https://leetcode.com/problems/find-score-of-an-array-after-marking-all-elements Title: 2593. Find Score of an Array After Marking All Elements Problem Description: You are given an array nums consisting of positive integers. Starting with score = 0, apply the following algorithm:

Choose the smallest integer of the array that is not marked. If there is a tie, choose the one with the smallest index. Add the value of the chosen integer to score. Mark the chosen element and its two adjacent elements if they exist. Repeat until all the array elements are marked.

Return the score you get after applying the above algorithm.   Example 1: Input: nums = [2,1,3,4,5,2] Output: 7 Explanation: We mark the elements as follows: - 1 is the smallest unmarked element, so we mark it and its two adjacent elements: [2,1,3,4,5,2]. - 2 is the smallest unmarked element, so we mark it and its left adjacent element: [2,1,3,4,5,2]. - 4 is the only remaining unmarked element, so we mark it: [2,1,3,4,5,2]. Our score is 1 + 2 + 4 = 7.

Example 2: Input: nums = [2,3,5,1,3,2] Output: 5 Explanation: We mark the elements as follows: - 1 is the smallest unmarked element, so we mark it and its two adjacent elements: [2,3,5,1,3,2]. - 2 is the smallest unmarked element, since there are two of them, we choose the left-most one, so we mark the one at index 0 and its right adjacent element: [2,3,5,1,3,2]. - 2 is the only remaining unmarked element, so we mark it: [2,3,5,1,3,2]. Our score is 1 + 2 + 2 = 5.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 106

==
Problem Number: 1588 URL: https://leetcode.com/problems/minimum-time-to-repair-cars Title: 2594. Minimum Time to Repair Cars Problem Description: You are given an integer array ranks representing the ranks of some mechanics. ranksi is the rank of the ith mechanic. A mechanic with a rank r can repair n cars in r * n2 minutes. You are also given an integer cars representing the total number of cars waiting in the garage to be repaired. Return the minimum

time taken to repair all the cars. Note: All the mechanics can repair the cars simultaneously. Example 1: Input: ranks = [4,2,3,1], cars = 10 Output: 16 Explanation: - The first mechanic will repair two cars. The time required is 4 * 2 * 2 = 16 minutes. - The second mechanic will repair two cars. The time required is 2 * 2 * 2 = 8 minutes. - The third mechanic will repair two cars. The time required is 3 * 2 * 2 = 12 minutes. - The fourth mechanic will repair four cars. The time required is 1 * 4 * 4 = 16 minutes. It can be proved that the cars cannot be repaired in less than 16 minutes.

Example 2: Input: ranks = [5,1,8], cars = 6 Output: 16 Explanation: - The first mechanic will repair one car. The time required is 5 * 1 * 1 = 5 minutes. - The second mechanic will repair four cars. The time required is 1 * 4 * 4 = 16 minutes. - The third mechanic will repair one car. The time required is 8 * 1 * 1 = 8 minutes. It can be proved that the cars cannot be repaired in less than 16 minutes.

Constraints:

1 <= ranks.length <= 105 1 <= ranks[i] <= 100 1 <= cars <= 106

==
Problem Number: 1589 URL: https://leetcode.com/problems/check-knight-tour-configuration Title: 2596. Check Knight Tour Configuration Problem Description: There is a knight on an n x n chessboard. In a valid configuration, the knight starts at the top-left cell of the board and visits every cell on the board exactly once. You are given an n x n integer matrix grid consisting of distinct integers from the range [0, n * n - 1] where grid[row][col] indicates that the cell (row, col) is the grid[row][col]th cell that the knight visited. The moves are 0-indexed. Return true if grid represents a valid configuration of the knight's movements or false otherwise. Note that a valid knight move consists of moving two squares vertically and one square horizontally, or two squares horizontally and one square vertically. The figure below illustrates all the possible eight moves of a knight from some cell.

Example 1:

Input: grid = [[0,11,16,5,20],[17,4,19,10,15],[12,1,8,21,6],[3,18,23,14,9],[24,13,2,7,22]] Output: true Explanation: The above diagram represents the grid. It can be shown that it is a valid configuration.

Example 2:

Input: grid = [[0,3,6],[5,8,1],[2,7,4]] Output: false Explanation: The above diagram represents the grid. The 8th move of the knight is not valid considering its position after the 7th move.

Constraints:

n == grid.length == grid[i].length 3 <= n <= 7 0 <= grid[row][col] < n * n All integers in grid are unique.

==
Problem Number: 1590 URL: https://leetcode.com/problems/the-number-of-beautiful-subsets Title: 2597. The Number of Beautiful Subsets Problem Description: You are given an array nums of positive integers and a positive integer k. A subset of nums is beautiful if it does not contain two integers with an absolute difference equal to k. Return the number of non-empty beautiful subsets of the array nums. A subset of nums is an array that can be obtained by deleting some (possibly none) elements from nums. Two subsets are different if and only if the chosen indices to delete are different. Example 1: Input: nums = [2,4,6], k = 2 Output: 4 Explanation: The beautiful subsets of the array nums are: [2], [4], [6], [2, 6]. It can be proved that there are only 4 beautiful subsets in the array [2,4,6].

Example 2: Input: nums = [1], k = 1 Output: 1 Explanation: The beautiful subset of the array nums is [1]. It can be proved that there is only 1 beautiful subset in the array [1].

Constraints:

1 <= nums.length <= 20 1 <= nums[i], k <= 1000

==
Problem Number: 1591 URL: https://leetcode.com/problems/smallest-missing-non-negative-integer-after-operations Title: 2598. Smallest Missing Non-negative Integer After Operations Problem Description: You are given a 0-indexed integer array nums and an integer value. In one operation, you can add or subtract value from any element of nums.

For example, if nums = [1,2,3] and value = 2, you can choose to subtract value from nums[0] to make nums = [-1,2,3].

The MEX (minimum excluded) of an array is the smallest missing non-negative integer in it.

For example, the MEX of [-1,2,3] is 0 while the MEX of [1,0,3] is 2.

Return the maximum MEX of nums after applying the mentioned operation any number of times. Example 1: Input: nums = [1,-10,7,13,6,8], value = 5 Output: 4 Explanation: One can achieve this result by applying the following operations: - Add value to nums[1] twice to make nums = [1,0,7,13,6,8] - Subtract value from nums[2] once to make nums = [1,0,2,13,6,8] - Subtract value from nums[3] twice to make nums = [1,0,2,3,6,8] The MEX of nums is 4. It can be shown that 4 is the maximum MEX we can achieve.

Example 2: Input: nums = [1,-10,7,13,6,8], value = 7 Output: 2 Explanation: One can achieve this result by applying the following operation: - subtract value from nums[2] once to make nums = [1,-10,0,13,6,8] The MEX of nums is 2. It can be shown that 2 is the maximum MEX we can achieve.

Constraints:

1 <= nums.length, value <= 105 -109 <= nums[i] <= 109

==

Problem Number: 1592 URL: https://leetcode.com/problems/prime-subtraction-operation Title: 2601. Prime Subtraction Operation Problem Description: You are given a 0-indexed integer array nums of length n. You can perform the following operation as many times as you want:

Pick an index i that you haven't picked before, and pick a prime p strictly less than nums[i], then subtract p from nums[i].

Return true if you can make nums a strictly increasing array using the above operation and false otherwise. A strictly increasing array is an array whose each element is strictly greater than its preceding element. Example 1: Input: nums = [4,9,6,10] Output: true Explanation: In the first operation: Pick i = 0 and p = 3, and then subtract 3 from nums[0], so that nums becomes [1,9,6,10]. In the second operation: i = 1, p = 7, subtract 7 from nums[1], so nums becomes equal to [1,2,6,10]. After the second operation, nums is sorted in strictly increasing order, so the answer is true. Example 2: Input: nums = [6,8,11,12] Output: true Explanation: Initially nums is sorted in strictly increasing order, so we don't need to make any operations. Example 3: Input: nums = [5,8,3] Output: false Explanation: It can be proven that there is no way to perform operations to make nums sorted in strictly increasing order, so the answer is false. Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 1000 nums.length == n

==

Problem Number: 1593 URL: https://leetcode.com/problems/minimum-operations-to-make-all-array-elements-equal Title: 2602. Minimum Operations to Make All Array Elements Equal Problem Description: You are given an array nums consisting of positive integers. You are also given an integer array queries of size m. For the ith query, you want to make all of the elements of nums equal to queries[i]. You can perform the following operation on the array any number of times:

Increase or decrease an element of the array by 1.

Return an array answer of size m where answer[i] is the minimum number of operations to make all elements of nums equal to queries[i]. Note that after each query the array is reset to its original state. Example 1: Input: nums = [3,1,6,8], queries = [1,5] Output: [14,10] Explanation: For the first query we can do the following operations: - Decrease nums[0] 2 times, so that nums = [1,1,6,8]. - Decrease nums[2] 5 times, so that nums = [1,1,1,8]. - Decrease nums[3] 7 times, so that nums = [1,1,1,1]. So the total number of operations for the first query is 2 + 5 + 7 = 14. For the second query we can do the following operations: - Increase nums[0] 2 times, so that nums = [5,1,6,8]. - Increase nums[1] 4 times, so that nums = [5,5,6,8]. - Decrease nums[2] 1 time, so that nums = [5,5,5,8]. - Decrease nums[3] 3 times, so that nums = [5,5,5,5]. So the total number of operations for the second query is 2 + 4 + 1 + 3 = 10.

Example 2: Input: nums = [2,9,6,3], queries = [10] Output: [20] Explanation: We can increase each value in the array to 10. The total number of operations will be 8 + 1 + 4 + 7 = 20.

Constraints:

n == nums.length m == queries.length 1 <= n, m <= 105 1 <= nums[i], queries[i] <= 109

==
Problem Number: 1594 URL: https://leetcode.com/problems/find-the-substring-with-maximum-cost Title: 2606. Find the Substring With Maximum Cost Problem Description: You are given a string s, a string chars of distinct characters and an integer array vals of the same length as chars. The cost of the substring is the sum of the values of each character in the substring. The cost of an empty string is considered 0. The value of the character is defined in the following way:

If the character is not in the string chars, then its value is its corresponding position (1-indexed) in the alphabet.

For example, the value of ’a’ is 1, the value of ’b’ is 2, and so on. The value of ’z’ is 26.

Otherwise, assuming i is the index where the character occurs in the string chars, then its value is vals[i].

Return the maximum cost among all substrings of the string s. Example 1: Input: s = ”adaa”, chars = ”d”, vals = [-1000] Output: 2 Explanation: The value of the characters ”a” and ”d” is 1 and -1000 respectively. The substring with the maximum cost is ”aa” and its cost is 1 + 1 = 2. It can be proven that 2 is the maximum cost.

Example 2: Input: s = ”abc”, chars = ”abc”, vals = [-1,-1,-1] Output: 0 Explanation: The value of the characters ”a”, ”b” and ”c” is -1, -1, and -1 respectively. The substring with the maximum cost is the empty substring ”” and its cost is 0. It can be proven that 0 is the maximum cost.

Constraints:

1 <= s.length <= 105 s consist of lowercase English letters. 1 <= chars.length <= 26 chars consist of distinct lowercase English letters. vals.length == chars.length -1000 <= vals[i] <= 1000

==
Problem Number: 1595 URL: https://leetcode.com/problems/make-k-subarray-sums-equal Title: 2607. Make K-Subarray Sums Equal Problem Description: You are given a 0-indexed integer array arr and an integer k. The array arr is circular. In other words, the first element of the array is the next element of

the last element, and the last element of the array is the previous element of the first element. You can do the following operation any number of times:

Pick any element from arr and increase or decrease it by 1.

Return the minimum number of operations such that the sum of each subarray of length k is equal. A subarray is a contiguous part of the array.   Example 1: Input: arr = [1,4,1,3], k = 2 Output: 1 Explanation: we can do one operation on index 1 to make its value equal to 3. The array after the operation is [1,3,1,3] - Subarray starts at index 0 is [1, 3], and its sum is 4 - Subarray starts at index 1 is [3, 1], and its sum is 4 - Subarray starts at index 2 is [1, 3], and its sum is 4 - Subarray starts at index 3 is [3, 1], and its sum is 4

Example 2: Input: arr = [2,5,5,7], k = 3 Output: 5 Explanation: we can do three operations on index 0 to make its value equal to 5 and two operations on index 3 to make its value equal to 5. The array after the operations is [5,5,5,5] - Subarray starts at index 0 is [5, 5, 5], and its sum is 15 - Subarray starts at index 1 is [5, 5, 5], and its sum is 15 - Subarray starts at index 2 is [5, 5, 5], and its sum is 15 - Subarray starts at index 3 is [5, 5, 5], and its sum is 15

Constraints:

1 <= k <= arr.length <= 105 1 <= arr[i] <= 109

==
Problem Number: 1596 URL: https://leetcode.com/problems/convert-an-array-into-a-2d-array-with-conditions Title: 2610. Convert an Array Into a 2D Array With Conditions Problem Description: You are given an integer array nums. You need to create a 2D array from nums satisfying the following conditions:

The 2D array should contain only the elements of the array nums. Each row in the 2D array contains distinct integers. The number of rows in the 2D array should be minimal.

Return the resulting array. If there are multiple answers, return any of them. Note that the 2D array can have a different number of elements on each row. Example 1: Input: nums = [1,3,4,1,2,3,1] Output: [[1,3,4,2],[1,3],[1]] Explanation: We can create a 2D array that contains the following rows: - 1,3,4,2 - 1,3 - 1 All elements of nums were used, and each row of the 2D array contains distinct integers, so it is a valid answer. It can be shown that we cannot have less than 3 rows in a valid array. Example 2: Input: nums = [1,2,3,4] Output: [[4,3,2,1]] Explanation: All elements of the array are distinct, so we can keep all of them in the first row of the 2D array.

Constraints:

1 <= nums.length <= 200 1 <= nums[i] <= nums.length

==
Problem Number: 1597 URL: https://leetcode.com/problems/mice-and-cheese Title: 2611. Mice and Cheese Problem Description: There are two mice and n

different types of cheese, each type of cheese should be eaten by exactly one mouse. A point of the cheese with index i (0-indexed) is:

reward1[i] if the first mouse eats it. reward2[i] if the second mouse eats it.

You are given a positive integer array reward1, a positive integer array reward2, and a non-negative integer k. Return the maximum points the mice can achieve if the first mouse eats exactly k types of cheese.   Example 1: Input: reward1 = [1,1,3,4], reward2 = [4,4,1,1], k = 2 Output: 15 Explanation: In this example, the first mouse eats the 2nd (0-indexed) and the 3rd types of cheese, and the second mouse eats the 0th and the 1st types of cheese. The total points are 4 + 4 + 3 + 4 = 15. It can be proven that 15 is the maximum total points that the mice can achieve.

Example 2: Input: reward1 = [1,1], reward2 = [1,1], k = 2 Output: 2 Explanation: In this example, the first mouse eats the 0th (0-indexed) and 1st types of cheese, and the second mouse does not eat any cheese. The total points are 1 + 1 = 2. It can be proven that 2 is the maximum total points that the mice can achieve.

Constraints:

1 <= n == reward1.length == reward2.length <= 105 1 <= reward1[i], reward2[i] <= 1000 0 <= k <= n

==
Problem Number: 1598 URL: https://leetcode.com/problems/sum-of-distances Title: 2615. Sum of Distances Problem Description: You are given a 0-indexed integer array nums. There exists an array arr of length nums.length, where arr[i] is the sum of |i - j| over all j such that nums[j] == nums[i] and j != i. If there is no such j, set arr[i] to be 0. Return the array arr.   Example 1: Input: nums = [1,3,1,1,2] Output: [5,0,3,4,0] Explanation: When i = 0, nums[0] == nums[2] and nums[0] == nums[3]. Therefore, arr[0] = |0 - 2| + |0 - 3| = 5. When i = 1, arr[1] = 0 because there is no other index with value 3. When i = 2, nums[2] == nums[0] and nums[2] == nums[3]. Therefore, arr[2] = |2 - 0| + |2 - 3| = 3. When i = 3, nums[3] == nums[0] and nums[3] == nums[2]. Therefore, arr[3] = |3 - 0| + |3 - 2| = 4. When i = 4, arr[4] = 0 because there is no other index with value 2.

Example 2: Input: nums = [0,5,3] Output: [0,0,0] Explanation: Since each element in nums is distinct, arr[i] = 0 for all i.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==
Problem Number: 1599 URL: https://leetcode.com/problems/minimize-the-maximum-difference-of-pairs Title: 2616. Minimize the Maximum Difference of Pairs Problem Description: You are given a 0-indexed integer array nums and an integer p. Find p pairs of indices of nums such that the maximum difference

amongst all the pairs is minimized. Also, ensure no index appears more than once amongst the p pairs. Note that for a pair of elements at the index i and j, the difference of this pair is |nums[i] - nums[j]|, where |x| represents the absolute value of x. Return the minimum maximum difference among all p pairs. We define the maximum of an empty set to be zero. Example 1: Input: nums = [10,1,2,7,1,3], p = 2 Output: 1 Explanation: The first pair is formed from the indices 1 and 4, and the second pair is formed from the indices 2 and 5. The maximum difference is max(|nums[1] - nums[4]|, |nums[2] - nums[5]|) = max(0, 1) = 1. Therefore, we return 1.

Example 2: Input: nums = [4,2,1,2], p = 1 Output: 0 Explanation: Let the indices 1 and 3 form a pair. The difference of that pair is |2 - 2| = 0, which is the minimum we can attain.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109 0 <= p <= (nums.length)/2

==
Problem Number: 1600 URL: https://leetcode.com/problems/find-the-score-of-all-prefixes-of-an-array Title: 2640. Find the Score of All Prefixes of an Array Problem Description: We define the conversion array conver of an array arr as follows:

conver[i] = arr[i] + max(arr[0..i]) where max(arr[0..i]) is the maximum value of arr[j] over 0 <= j <= i.

We also define the score of an array arr as the sum of the values of the conversion array of arr. Given a 0-indexed integer array nums of length n, return an array ans of length n where ans[i] is the score of the prefix nums[0..i]. Example 1: Input: nums = [2,3,7,5,10] Output: [4,10,24,36,56] Explanation: For the prefix [2], the conversion array is [4] hence the score is 4 For the prefix [2, 3], the conversion array is [4, 6] hence the score is 10 For the prefix [2, 3, 7], the conversion array is [4, 6, 14] hence the score is 24 For the prefix [2, 3, 7, 5], the conversion array is [4, 6, 14, 12] hence the score is 36 For the prefix [2, 3, 7, 5, 10], the conversion array is [4, 6, 14, 12, 20] hence the score is 56

Example 2: Input: nums = [1,1,2,4,8,16] Output: [2,4,8,16,32,64] Explanation: For the prefix [1], the conversion array is [2] hence the score is 2 For the prefix [1, 1], the conversion array is [2, 2] hence the score is 4 For the prefix [1, 1, 2], the conversion array is [2, 2, 4] hence the score is 8 For the prefix [1, 1, 2, 4], the conversion array is [2, 2, 4, 8] hence the score is 16 For the prefix [1, 1, 2, 4, 8], the conversion array is [2, 2, 4, 8, 16] hence the score is 32 For the prefix [1, 1, 2, 4, 8, 16], the conversion array is [2, 2, 4, 8, 16, 32] hence the score is 64

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1601 URL: https://leetcode.com/problems/cousins-in-

binary-tree-ii Title: 2641. Cousins in Binary Tree II Problem Description: Given the root of a binary tree, replace the value of each node in the tree with the sum of all its cousins' values. Two nodes of a binary tree are cousins if they have the same depth with different parents. Return the root of the modified tree. Note that the depth of a node is the number of edges in the path from the root node to it.   Example 1:

Input: root = [5,4,9,1,10,null,7] Output: [0,0,0,7,7,null,11] Explanation: The diagram above shows the initial binary tree and the binary tree after changing the value of each node. - Node with value 5 does not have any cousins so its sum is 0. - Node with value 4 does not have any cousins so its sum is 0. - Node with value 9 does not have any cousins so its sum is 0. - Node with value 1 has a cousin with value 7 so its sum is 7. - Node with value 10 has a cousin with value 7 so its sum is 7. - Node with value 7 has cousins with values 1 and 10 so its sum is 11.

Example 2:

Input: root = [3,1,2] Output: [0,0,0] Explanation: The diagram above shows the initial binary tree and the binary tree after changing the value of each node. - Node with value 3 does not have any cousins so its sum is 0. - Node with value 1 does not have any cousins so its sum is 0. - Node with value 2 does not have any cousins so its sum is 0.

Constraints:

The number of nodes in the tree is in the range [1, 105]. 1 <= Node.val <= 104

==

Problem Number: 1602 URL: https://leetcode.com/problems/minimum-additions-to-make-valid-string Title: 2645. Minimum Additions to Make Valid String Problem Description: Given a string word to which you can insert letters "a", "b" or "c" anywhere and any number of times, return the minimum number of letters that must be inserted so that word becomes valid. A string is called valid if it can be formed by concatenating the string "abc" several times. Example 1: Input: word = "b" Output: 2 Explanation: Insert the letter "a" right before "b", and the letter "c" right next to "a" to obtain the valid string "abc".

Example 2: Input: word = "aaa" Output: 6 Explanation: Insert letters "b" and "c" next to each "a" to obtain the valid string "abcabcabc".

Example 3: Input: word = "abc" Output: 0 Explanation: word is already valid. No modifications are needed.

Constraints:

1 <= word.length <= 50 word consists of letters "a", "b" and "c" only.

==

Problem Number: 1603 URL: https://leetcode.com/problems/sliding-subarray-beauty Title: 2653. Sliding Subarray Beauty Problem Description: Given an integer array nums containing n integers, find the beauty of each subarray of size k. The beauty of a subarray is the xth smallest integer in the subarray if it is negative, or 0 if there are fewer than x negative integers. Return an integer array containing n - k + 1 integers, which denote the beauty of the subarrays in order from the first index in the array.

A subarray is a contiguous non-empty sequence of elements within an array.

Example 1: Input: nums = [1,-1,-3,-2,3], k = 3, x = 2 Output: [-1,-2,-2] Explanation: There are 3 subarrays with size k = 3. The first subarray is [1, -1, -3] and the 2nd smallest negative integer is -1. The second subarray is [-1, -3, -2] and the 2nd smallest negative integer is -2. The third subarray is [-3, -2, 3] and the 2nd smallest negative integer is -2. Example 2: Input: nums = [-1,-2,-3,-4,-5], k = 2, x = 2 Output: [-1,-2,-3,-4] Explanation: There are 4 subarrays with size k = 2. For [-1, -2], the 2nd smallest negative integer is -1. For [-2, -3], the 2nd smallest negative integer is -2. For [-3, -4], the 2nd smallest negative integer is -3. For [-4, -5], the 2nd smallest negative integer is -4. Example 3: Input: nums = [-3,1,2,-3,0,-3], k = 2, x = 1 Output: [-3,0,-3,-3,-3] Explanation: There are 5 subarrays with size k = 2. For [-3, 1], the 1st smallest negative integer is -3. For [1, 2], there is no negative integer so the beauty is 0. For [2, -3], the 1st smallest negative integer is -3. For [-3, 0], the 1st smallest negative integer is -3. For [0, -3], the 1st smallest negative integer is -3. Constraints:

n == nums.length 1 <= n <= 105 1 <= k <= n 1 <= x <= k -50 <= nums[i] <= 50

==

Problem Number: 1604 URL: https://leetcode.com/problems/minimum-number-of-operations-to-make-all-array-elements-equal-to-1 Title: 2654. Minimum Number of Operations to Make All Array Elements Equal to 1 Problem Description: You are given a 0-indexed array nums consisiting of positive integers. You can do the following operation on the array any number of times:

Select an index i such that 0 <= i < n - 1 and replace either of nums[i] or nums[i+1] with their gcd value.

Return the minimum number of operations to make all elements of nums equal to 1. If it is impossible, return -1. The gcd of two integers is the greatest common divisor of the two integers. Example 1: Input: nums = [2,6,3,4] Output: 4 Explanation: We can do the following operations: - Choose index i = 2 and replace nums[2] with gcd(3,4) = 1. Now we have nums = [2,6,1,4]. - Choose index i = 1 and replace nums[1] with gcd(6,1) = 1. Now we have nums = [2,1,1,4]. - Choose index i = 0 and replace nums[0] with gcd(2,1) = 1. Now we have nums = [1,1,1,4]. - Choose index i = 2 and replace nums[3] with gcd(1,4) = 1. Now we have nums = [1,1,1,1].

Example 2: Input: nums = [2,10,6,14] Output: -1 Explanation: It can be shown that it is impossible to make all the elements equal to 1.

Constraints:

2 <= nums.length <= 50 1 <= nums[i] <= 106

Follow-up: The O(n) time complexity solution works, but could you find an O(1) constant time complexity solution?

===

Problem Number: 1605 URL: https://leetcode.com/problems/find-the-prefix-common-array-of-two-arrays Title: 2657. Find the Prefix Common Array of Two Arrays Problem Description: You are given two 0-indexed integer permutations A and B of length n. A prefix common array of A and B is an array C such that C[i] is equal to the count of numbers that are present at or before the index i in both A and B. Return the prefix common array of A and B. A sequence of n integers is called a permutation if it contains all integers from 1 to n exactly once. Example 1: Input: A = [1,3,2,4], B = [3,1,2,4] Output: [0,2,3,4] Explanation: At i = 0: no number is common, so C[0] = 0. At i = 1: 1 and 3 are common in A and B, so C[1] = 2. At i = 2: 1, 2, and 3 are common in A and B, so C[2] = 3. At i = 3: 1, 2, 3, and 4 are common in A and B, so C[3] = 4.

Example 2: Input: A = [2,3,1], B = [3,1,2] Output: [0,1,3] Explanation: At i = 0: no number is common, so C[0] = 0. At i = 1: only 3 is common in A and B, so C[1] = 1. At i = 2: 1, 2, and 3 are common in A and B, so C[2] = 3.

Constraints:

1 <= A.length == B.length == n <= 50 1 <= A[i], B[i] <= n It is guaranteed that A and B are both a permutation of n integers.

===

Problem Number: 1606 URL: https://leetcode.com/problems/maximum-number-of-fish-in-a-grid Title: 2658. Maximum Number of Fish in a Grid Problem Description: You are given a 0-indexed 2D matrix grid of size m x n, where (r, c) represents:

A land cell if grid[r][c] = 0, or A water cell containing grid[r][c] fish, if grid[r][c] > 0.

A fisher can start at any water cell (r, c) and can do the following operations any number of times:

Catch all the fish at cell (r, c), or Move to any adjacent water cell.

Return the maximum number of fish the fisher can catch if he chooses his starting cell optimally, or 0 if no water cell exists. An adjacent cell of the cell (r, c), is one of the cells (r, c + 1), (r, c - 1), (r + 1, c) or (r - 1, c) if it exists. Example 1:

Input: grid = [[0,2,1,0],[4,0,0,3],[1,0,0,4],[0,3,2,0]] Output: 7 Explanation: The fisher can start at cell (1,3) and collect 3 fish, then move to cell (2,3) and collect 4 fish.

Example 2:

Input: grid = [[1,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,1]] Output: 1 Explanation: The fisher can start at cells (0,0) or (3,3) and collect a single fish.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 10 0 <= grid[i][j] <= 10

==

Problem Number: 1607 URL: https://leetcode.com/problems/first-completely-painted-row-or-column Title: 2661. First Completely Painted Row or Column Problem Description: You are given a 0-indexed integer array arr, and an m x n integer matrix mat. arr and mat both contain all the integers in the range [1, m * n]. Go through each index i in arr starting from index 0 and paint the cell in mat containing the integer arr[i]. Return the smallest index i at which either a row or a column will be completely painted in mat. Example 1:

Input: arr = [1,3,4,2], mat = [[1,4],[2,3]] Output: 2 Explanation: The moves are shown in order, and both the first row and second column of the matrix become fully painted at arr[2].

Example 2:

Input: arr = [2,8,7,4,1,3,5,6,9], mat = [[3,2,5],[1,4,6],[8,7,9]] Output: 3 Explanation: The second column becomes fully painted at arr[3].

Constraints:

m == mat.length n = mat[i].length arr.length == m * n 1 <= m, n <= 105 1 <= m * n <= 105 1 <= arr[i], mat[r][c] <= m * n All the integers of arr are unique. All the integers of mat are unique.

==

Problem Number: 1608 URL: https://leetcode.com/problems/minimum-cost-of-a-path-with-special-roads Title: 2662. Minimum Cost of a Path With Special Roads Problem Description: You are given an array start where start = [startX, startY] represents your initial position (startX, startY) in a 2D space. You are also given the array target where target = [targetX, targetY] represents your target position (targetX, targetY). The cost of going from a position (x1, y1) to any other position in the space (x2, y2) is |x2 - x1| + |y2 - y1|. There are also some special roads. You are given a 2D array specialRoads where specialRoads[i] = [x1i, y1i, x2i, y2i, costi] indicates that the ith special road can take you from (x1i, y1i) to (x2i, y2i) with a cost equal to costi. You can use each special road any number of times. Return the minimum cost required to go from (startX, startY) to (targetX, targetY). Example 1: Input: start = [1,1], target = [4,5], specialRoads = [[1,2,3,3,2],[3,4,4,5,1]] Output: 5

Explanation: The optimal path from (1,1) to (4,5) is the following: - (1,1) -> (1,2). This move has a cost of |1 - 1| + |2 - 1| = 1. - (1,2) -> (3,3). This move uses the first special edge, the cost is 2. - (3,3) -> (3,4). This move has a cost of |3 - 3| + |4 - 3| = 1. - (3,4) -> (4,5). This move uses the second special edge, the cost is 1. So the total cost is 1 + 2 + 1 + 1 = 5. It can be shown that we cannot achieve a smaller total cost than 5.

Example 2: Input: start = [3,2], target = [5,7], specialRoads = [[3,2,3,4,4],[3,3,5,5,5],[3,4,5,6,6]] Output: 7 Explanation: It is optimal to not use any special edges and go directly from the starting to the ending position with a cost |5 - 3| + |7 - 2| = 7.

Constraints:

start.length == target.length == 2 1 <= startX <= targetX <= 105 1 <= startY <= targetY <= 105 1 <= specialRoads.length <= 200 specialRoads[i].length == 5 startX <= x1i, x2i <= targetX startY <= y1i, y2i <= targetY 1 <= costi <= 105

==
Problem Number: 1609 URL: https://leetcode.com/problems/frequency-tracker Title: 2671. Frequency Tracker Problem Description: Design a data structure that keeps track of the values in it and answers some queries regarding their frequencies. Implement the FrequencyTracker class.

FrequencyTracker(): Initializes the FrequencyTracker object with an empty array initially. void add(int number): Adds number to the data structure. void deleteOne(int number): Deletes one occurrence of number from the data structure. The data structure may not contain number, and in this case nothing is deleted. bool hasFrequency(int frequency): Returns true if there is a number in the data structure that occurs frequency number of times, otherwise, it returns false.

Example 1: Input ["FrequencyTracker", "add", "add", "hasFrequency"] [[], [3], [3], [2]] Output [null, null, null, true]

Explanation FrequencyTracker frequencyTracker = new FrequencyTracker(); frequencyTracker.add(3); // The data structure now contains [3] frequencyTracker.add(3); // The data structure now contains [3, 3] frequencyTracker.hasFrequency(2); // Returns true, because 3 occurs twice

Example 2: Input ["FrequencyTracker", "add", "deleteOne", "hasFrequency"] [[], [1], [1], [1]] Output [null, null, null, false]

Explanation FrequencyTracker frequencyTracker = new FrequencyTracker(); frequencyTracker.add(1); // The data structure now contains [1] frequencyTracker.deleteOne(1); // The data structure becomes empty [] frequencyTracker.hasFrequency(1); // Returns false, because the data structure is empty

Example 3: Input ["FrequencyTracker", "hasFrequency", "add", "hasFrequency"] [[], [2], [3], [1]] Output [null, false, null, true]

Explanation FrequencyTracker frequencyTracker = new FrequencyTracker(); frequencyTracker.hasFrequency(2); // Returns false, because the data structure is empty frequencyTracker.add(3); // The data structure now contains [3] frequencyTracker.hasFrequency(1); // Returns true, because 3 occurs once

Constraints:

1 <= number <= 105 1 <= frequency <= 105 At most, 2 * 105 calls will be made to add, deleteOne, and hasFrequency in total.

==
Problem Number: 1610 URL: https://leetcode.com/problems/number-of-adjacent-elements-with-the-same-color Title: 2672. Number of Adjacent Elements With the Same Color Problem Description: There is a 0-indexed array nums of length n. Initially, all elements are uncolored (has a value of 0). You are given a 2D integer array queries where queries[i] = [indexi, colori]. For each query, you color the index indexi with the color colori in the array nums. Return an array answer of the same length as queries where answer[i] is the number of adjacent elements with the same color after the ith query. More formally, answer[i] is the number of indices j, such that 0 <= j < n - 1 and nums[j] == nums[j + 1] and nums[j] != 0 after the ith query. Example 1: Input: n = 4, queries = [[0,2],[1,2],[3,1],[1,1],[2,1]] Output: [0,1,1,0,2] Explanation: Initially array nums = [0,0,0,0], where 0 denotes uncolored elements of the array. - After the 1st query nums = [2,0,0,0]. The count of adjacent elements with the same color is 0. - After the 2nd query nums = [2,2,0,0]. The count of adjacent elements with the same color is 1. - After the 3rd query nums = [2,2,0,1]. The count of adjacent elements with the same color is 1. - After the 4th query nums = [2,1,0,1]. The count of adjacent elements with the same color is 0. - After the 5th query nums = [2,1,1,1]. The count of adjacent elements with the same color is 2.

Example 2: Input: n = 1, queries = [[0,100000]] Output: [0] Explanation: Initially array nums = [0], where 0 denotes uncolored elements of the array. - After the 1st query nums = [100000]. The count of adjacent elements with the same color is 0.

Constraints:

1 <= n <= 105 1 <= queries.length <= 105 queries[i].length == 2 0 <= indexi <= n - 1 1 <= colori <= 105

==
Problem Number: 1611 URL: https://leetcode.com/problems/make-costs-of-paths-equal-in-a-binary-tree Title: 2673. Make Costs of Paths Equal in a Binary Tree Problem Description: You are given an integer n representing the number of nodes in a perfect binary tree consisting of nodes numbered from 1 to n. The root of the tree is node 1 and each node i in the tree has two children

where the left child is the node 2 * i and the right child is 2 * i + 1. Each node in the tree also has a cost represented by a given 0-indexed integer array cost of size n where cost[i] is the cost of node i + 1. You are allowed to increment the cost of any node by 1 any number of times. Return the minimum number of increments you need to make the cost of paths from the root to each leaf node equal. Note:

A perfect binary tree is a tree where each node, except the leaf nodes, has exactly 2 children. The cost of a path is the sum of costs of nodes in the path.

Example 1:

Input: n = 7, cost = [1,5,2,2,3,3,1] Output: 6 Explanation: We can do the following increments: - Increase the cost of node 4 one time. - Increase the cost of node 3 three times. - Increase the cost of node 7 two times. Each path from the root to a leaf will have a total cost of 9. The total increments we did is 1 + 3 + 2 = 6. It can be shown that this is the minimum answer we can achieve.

Example 2:

Input: n = 3, cost = [5,3,3] Output: 0 Explanation: The two paths already have equal total costs, so no increments are needed.

Constraints:

3 <= n <= 105 n + 1 is a power of 2 cost.length == n 1 <= cost[i] <= 104

==
Problem Number: 1612 URL: https://leetcode.com/problems/sum-in-a-matrix Title: 2679. Sum in a Matrix Problem Description: You are given a 0-indexed 2D integer array nums. Initially, your score is 0. Perform the following operations until the matrix becomes empty:

From each row in the matrix, select the largest number and remove it. In the case of a tie, it does not matter which number is chosen. Identify the highest number amongst all those removed in step 1. Add that number to your score.

Return the final score. Example 1: Input: nums = [[7,2,1],[6,4,2],[6,5,3],[3,2,1]] Output: 15 Explanation: In the first operation, we remove 7, 6, 6, and 3. We then add 7 to our score. Next, we remove 2, 4, 5, and 2. We add 5 to our score. Lastly, we remove 1, 2, 3, and 1. We add 3 to our score. Thus, our final score is 7 + 5 + 3 = 15.

Example 2: Input: nums = [[1]] Output: 1 Explanation: We remove 1 and add it to the answer. We return 1. Constraints:

1 <= nums.length <= 300 1 <= nums[i].length <= 500 0 <= nums[i][j] <= 103

==
Problem Number: 1613 URL: https://leetcode.com/problems/maximum-or Title: 2680. Maximum OR Problem Description: You are given a 0-indexed

integer array nums of length n and an integer k. In an operation, you can choose an element and multiply it by 2. Return the maximum possible value of nums[0] | nums[1] | ... | nums[n - 1] that can be obtained after applying the operation on nums at most k times. Note that a | b denotes the bitwise or between two integers a and b.   Example 1: Input: nums = [12,9], k = 1 Output: 30 Explanation: If we apply the operation to index 1, our new array nums will be equal to [12,18]. Thus, we return the bitwise or of 12 and 18, which is 30.

Example 2: Input: nums = [8,1,2], k = 2 Output: 35 Explanation: If we apply the operation twice on index 0, we yield a new array of [32,1,2]. Thus, we return 32|1|2 = 35.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= k <= 15

==

Problem Number: 1614 URL: https://leetcode.com/problems/neighboring-bitwise-xor Title: 2683. Neighboring Bitwise XOR Problem Description: A 0-indexed array derived with length n is derived by computing the bitwise XOR ( ) of adjacent values in a binary array original of length n. Specifically, for each index i in the range [0, n - 1]:

If i = n - 1, then derived[i] = original[i] original[0]. Otherwise, derived[i] = original[i] original[i + 1].

Given an array derived, your task is to determine whether there exists a valid binary array original that could have formed derived. Return true if such an array exists or false otherwise.

A binary array is an array containing only 0's and 1's

Example 1: Input: derived = [1,1,0] Output: true Explanation: A valid original array that gives derived is [0,1,0]. derived[0] = original[0] original[1] = 0 1 = 1 derived[1] = original[1] original[2] = 1 0 = 1 derived[2] = original[2] original[0] = 0 0 = 0

Example 2: Input: derived = [1,1] Output: true Explanation: A valid original array that gives derived is [0,1]. derived[0] = original[0] original[1] = 1 derived[1] = original[1] original[0] = 1

Example 3: Input: derived = [1,0] Output: false Explanation: There is no valid original array that gives derived.

Constraints:

n == derived.length 1 <= n <= 105 The values in derived are either 0's or 1's

==

Problem Number: 1615 URL: https://leetcode.com/problems/maximum-number-of-moves-in-a-grid Title: 2684. Maximum Number of Moves in a Grid

Problem Description: You are given a 0-indexed m x n matrix grid consisting of positive integers. You can start at any cell in the first column of the matrix, and traverse the grid in the following way:

From a cell (row, col), you can move to any of the cells: (row - 1, col + 1), (row, col + 1) and (row + 1, col + 1) such that the value of the cell you move to, should be strictly bigger than the value of the current cell.

Return the maximum number of moves that you can perform.   Example 1:

Input: grid = [[2,4,3,5],[5,4,9,3],[3,4,2,11],[10,9,13,15]] Output: 3 Explanation: We can start at the cell (0, 0) and make the following moves: - (0, 0) -> (0, 1). - (0, 1) -> (1, 2). - (1, 2) -> (2, 3). It can be shown that it is the maximum number of moves that can be made. Example 2:

Input: grid = [[3,2,4],[2,1,9],[1,1,7]] Output: 0 Explanation: Starting from any cell in the first column we cannot perform any moves.

Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 1000 4 <= m * n <= 105 1 <= grid[i][j] <= 106

==
Problem Number: 1616 URL: https://leetcode.com/problems/count-the-number-of-complete-components Title: 2685. Count the Number of Complete Components Problem Description: You are given an integer n. There is an undirected graph with n vertices, numbered from 0 to n - 1. You are given a 2D integer array edges where edges[i] = [ai, bi] denotes that there exists an undirected edge connecting vertices ai and bi. Return the number of complete connected components of the graph. A connected component is a subgraph of a graph in which there exists a path between any two vertices, and no vertex of the subgraph shares an edge with a vertex outside of the subgraph. A connected component is said to be complete if there exists an edge between every pair of its vertices.   Example 1:

Input: n = 6, edges = [[0,1],[0,2],[1,2],[3,4]] Output: 3 Explanation: From the picture above, one can see that all of the components of this graph are complete.

Example 2:

Input: n = 6, edges = [[0,1],[0,2],[1,2],[3,4],[3,5]] Output: 1 Explanation: The component containing vertices 0, 1, and 2 is complete since there is an edge between every pair of two vertices. On the other hand, the component containing vertices 3, 4, and 5 is not complete since there is no edge between vertices 4 and 5. Thus, the number of complete components in this graph is 1.

Constraints:

1 <= n <= 50 0 <= edges.length <= n * (n - 1) / 2 edges[i].length == 2 0 <= ai, bi <= n - 1 ai != bi There are no repeated edges.

==
Problem Number: 1617 URL: https://leetcode.com/problems/find-the-punishment-number-of-an-integer Title: 2698. Find the Punishment Number of an Integer Problem Description: Given a positive integer n, return the punishment number of n. The punishment number of n is defined as the sum of the squares of all integers i such that:

1 <= i <= n The decimal representation of i * i can be partitioned into contiguous substrings such that the sum of the integer values of these substrings equals i.

Example 1: Input: n = 10 Output: 182 Explanation: There are exactly 3 integers i that satisfy the conditions in the statement: - 1 since 1 * 1 = 1 - 9 since 9 * 9 = 81 and 81 can be partitioned into 8 + 1. - 10 since 10 * 10 = 100 and 100 can be partitioned into 10 + 0. Hence, the punishment number of 10 is 1 + 81 + 100 = 182

Example 2: Input: n = 37 Output: 1478 Explanation: There are exactly 4 integers i that satisfy the conditions in the statement: - 1 since 1 * 1 = 1. - 9 since 9 * 9 = 81 and 81 can be partitioned into 8 + 1. - 10 since 10 * 10 = 100 and 100 can be partitioned into 10 + 0. - 36 since 36 * 36 = 1296 and 1296 can be partitioned into 1 + 29 + 6. Hence, the punishment number of 37 is 1 + 81 + 100 + 1296 = 1478

Constraints:

1 <= n <= 1000

==
Problem Number: 1618 URL: https://leetcode.com/problems/extra-characters-in-a-string Title: 2707. Extra Characters in a String Problem Description: You are given a 0-indexed string s and a dictionary of words dictionary. You have to break s into one or more non-overlapping substrings such that each substring is present in dictionary. There may be some extra characters in s which are not present in any of the substrings. Return the minimum number of extra characters left over if you break up s optimally. Example 1: Input: s = "leetscode", dictionary = ["leet","code","leetcode"] Output: 1 Explanation: We can break s in two substrings: "leet" from index 0 to 3 and "code" from index 5 to 8. There is only 1 unused character (at index 4), so we return 1.

Example 2: Input: s = "sayhelloworld", dictionary = ["hello","world"] Output: 3 Explanation: We can break s in two substrings: "hello" from index 3 to 7 and "world" from index 8 to 12. The characters at indices 0, 1, 2 are not used in any substring and thus are considered as extra characters. Hence, we return 3.

Constraints:

1 <= s.length <= 50 1 <= dictionary.length <= 50 1 <= dictionary[i].length <= 50 dictionary[i] and s consists of only lowercase English letters dictionary contains distinct words

== Problem Number: 1619 URL: https://leetcode.com/problems/maximum-strength-of-a-group Title: 2708. Maximum Strength of a Group Problem Description: You are given a 0-indexed integer array nums representing the score of students in an exam. The teacher would like to form one non-empty group of students with maximal strength, where the strength of a group of students of indices i0, i1, i2, ... , ik is defined as nums[i0] * nums[i1] * nums[i2] * ... * nums[ik]. Return the maximum strength of a group the teacher can create. Example 1: Input: nums = [3,-1,-5,2,5,-9] Output: 1350 Explanation: One way to form a group of maximal strength is to group the students at indices [0,2,3,4,5]. Their strength is 3 * (-5) * 2 * 5 * (-9) = 1350, which we can show is optimal.

Example 2: Input: nums = [-4,-5,-4] Output: 20 Explanation: Group the students at indices [0, 1] . Then, we'll have a resulting strength of 20. We cannot achieve greater strength.

Constraints:

1 <= nums.length <= 13 -9 <= nums[i] <= 9

== Problem Number: 1620 URL: https://leetcode.com/problems/difference-of-number-of-distinct-values-on-diagonals Title: 2711. Difference of Number of Distinct Values on Diagonals Problem Description: Given a 0-indexed 2D grid of size m x n, you should find the matrix answer of size m x n. The value of each cell (r, c) of the matrix answer is calculated in the following way:

Let topLeft[r][c] be the number of distinct values in the top-left diagonal of the cell (r, c) in the matrix grid. Let bottomRight[r][c] be the number of distinct values in the bottom-right diagonal of the cell (r, c) in the matrix grid.

Then answer[r][c] = |topLeft[r][c] - bottomRight[r][c]|. Return the matrix answer. A matrix diagonal is a diagonal line of cells starting from some cell in either the topmost row or leftmost column and going in the bottom-right direction until reaching the matrix's end. A cell (r1, c1) belongs to the top-left diagonal of the cell (r, c), if both belong to the same diagonal and r1 < r. Similarly is defined bottom-right diagonal. Example 1:

Input: grid = [[1,2,3],[3,1,5],[3,2,1]] Output: [[1,1,0],[1,0,1],[0,1,1]] Explanation: The 1st diagram denotes the initial grid. The 2nd diagram denotes a grid for cell (0,0), where blue-colored cells are cells on its bottom-right diagonal. The 3rd diagram denotes a grid for cell (1,2), where red-colored cells are cells on its top-left diagonal. The 4th diagram denotes a grid for cell (1,1), where blue-colored cells are cells on its bottom-right diagonal and red-colored cells are cells on its top-left diagonal. - The cell (0,0) contains [1,1] on its bottom-right diagonal and [] on its top-left diagonal. The answer is |1 - 0| = 1. - The cell (1,2) contains [] on its bottom-right diagonal and [2] on its top-left diagonal. The answer is |0 - 1| = 1. - The cell (1,1) contains [1] on its bottom-right diagonal and [1] on its top-left diagonal. The answer is |1 - 1| = 0. The answers of other cells are similarly calculated.

Example 2: Input: grid = [[1]] Output: [[0]] Explanation: - The cell (0,0) contains [] on its bottom-right diagonal and [] on its top-left diagonal. The answer is |0 - 0| = 0.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n, grid[i][j] <= 50

==

Problem Number: 1621 URL: https://leetcode.com/problems/minimum-cost-to-make-all-characters-equal Title: 2712. Minimum Cost to Make All Characters Equal Problem Description: You are given a 0-indexed binary string s of length n on which you can apply two types of operations:

Choose an index i and invert all characters from index 0 to index i (both inclusive), with a cost of i + 1 Choose an index i and invert all characters from index i to index n - 1 (both inclusive), with a cost of n - i

Return the minimum cost to make all characters of the string equal. Invert a character means if its value is '0' it becomes '1' and vice-versa. Example 1: Input: s = "0011" Output: 2 Explanation: Apply the second operation with i = 2 to obtain s = "0000" for a cost of 2. It can be shown that 2 is the minimum cost to make all characters equal.

Example 2: Input: s = "010101" Output: 9 Explanation: Apply the first operation with i = 2 to obtain s = "101101" for a cost of 3. Apply the first operation with i = 1 to obtain s = "011101" for a cost of 2. Apply the first operation with i = 0 to obtain s = "111101" for a cost of 1. Apply the second operation with i = 4 to obtain s = "111110" for a cost of 2. Apply the second operation with i = 5 to obtain s = "111111" for a cost of 1. The total cost to make all characters equal is 9. It can be shown that 9 is the minimum cost to make all characters equal.

Constraints:

1 <= s.length == n <= 105 s[i] is either '0' or '1'

==

Problem Number: 1622 URL: https://leetcode.com/problems/sum-of-matrix-after-queries Title: 2718. Sum of Matrix After Queries Problem Description: You are given an integer n and a 0-indexed 2D array queries where queries[i] = [typei, indexi, vali]. Initially, there is a 0-indexed n x n matrix filled with 0's. For each query, you must apply one of the following changes:

if typei == 0, set the values in the row with indexi to vali, overwriting any previous values. if typei == 1, set the values in the column with indexi to vali, overwriting any previous values.

Return the sum of integers in the matrix after all queries are applied. Example 1:

Input: n = 3, queries = [[0,0,1],[1,2,2],[0,2,3],[1,0,4]] Output: 23 Explanation: The image above describes the matrix after each query. The sum of the matrix after all queries are applied is 23.

Example 2:

Input: n = 3, queries = [[0,0,4],[0,1,2],[1,0,1],[0,2,3],[1,2,1]] Output: 17 Explanation: The image above describes the matrix after each query. The sum of the matrix after all queries are applied is 17.

Constraints:

1 <= n <= 104 1 <= queries.length <= 5 * 104 queries[i].length == 3 0 <= typei <= 1 0 <= indexi < n 0 <= vali <= 105

==

Problem Number: 1623 URL: https://leetcode.com/problems/find-the-longest-semi-repetitive-substring Title: 2730. Find the Longest Semi-Repetitive Substring Problem Description: You are given a 0-indexed string s that consists of digits from 0 to 9. A string t is called a semi-repetitive if there is at most one consecutive pair of the same digits inside t. For example, 0010, 002020, 0123, 2002, and 54944 are semi-repetitive while 00101022, and 1101234883 are not. Return the length of the longest semi-repetitive substring inside s. A substring is a contiguous non-empty sequence of characters within a string. Example 1: Input: s = "52233" Output: 4 Explanation: The longest semi-repetitive substring is "5223", which starts at i = 0 and ends at j = 3.

Example 2: Input: s = "5494" Output: 4 Explanation: s is a semi-reptitive string, so the answer is 4.

Example 3: Input: s = "1111111" Output: 2 Explanation: The longest semi-repetitive substring is "11", which starts at i = 0 and ends at j = 1.

Constraints:

1 <= s.length <= 50 '0' <= s[i] <= '9'

==

Problem Number: 1624 URL: https://leetcode.com/problems/movement-of-robots Title: 2731. Movement of Robots Problem Description: Some robots are standing on an infinite number line with their initial coordinates given by a 0-indexed integer array nums and will start moving once given the command to move. The robots will move a unit distance each second. You are given a string s denoting the direction in which robots will move on command. 'L' means the robot will move towards the left side or negative side of the number line, whereas 'R' means the robot will move towards the right side or positive side of the number line. If two robots collide, they will start moving in opposite directions. Return the sum of distances between all the pairs of robots d seconds after the command. Since the sum can be very large, return it modulo 109 + 7. Note:

For two robots at the index i and j, pair (i,j) and pair (j,i) are considered the same pair. When robots collide, they instantly change their directions without wasting any time. Collision happens when two robots share the same place in a moment.

For example, if a robot is positioned in 0 going to the right and another is positioned in 2 going to the left, the next second they'll be both in 1 and they will change direction and the next second the first one will be in 0, heading left, and another will be in 2, heading right. For example, if a robot is positioned in 0 going to the right and another is positioned in 1 going to the left, the next second the first one will be in 0, heading left, and another will be in 1, heading right.

Example 1: Input: nums = [-2,0,2], s = "RLL", d = 3 Output: 8 Explanation: After 1 second, the positions are [-1,-1,1]. Now, the robot at index 0 will move left, and the robot at index 1 will move right. After 2 seconds, the positions are [-2,0,0]. Now, the robot at index 1 will move left, and the robot at index 2 will move right. After 3 seconds, the positions are [-3,-1,1]. The distance between the robot at index 0 and 1 is abs(-3 - (-1)) = 2. The distance between the robot at index 0 and 2 is abs(-3 - 1) = 4. The distance between the robot at index 1 and 2 is abs(-1 - 1) = 2. The sum of the pairs of all distances = 2 + 4 + 2 = 8.

Example 2: Input: nums = [1,0], s = "RL", d = 2 Output: 5 Explanation: After 1 second, the positions are [2,-1]. After 2 seconds, the positions are [3,-2]. The distance between the two robots is abs(-2 - 3) = 5.

Constraints:

2 <= nums.length <= 105 -2 * 109 <= nums[i] <= 2 * 109 0 <= d <= 109 nums.length == s.length s consists of 'L' and 'R' only nums[i] will be unique.

==
Problem Number: 1625 URL: https://leetcode.com/problems/lexicographically-smallest-string-after-substring-operation Title: 2734. Lexicographically Smallest String After Substring Operation Problem Description: You are given a string s consisting of only lowercase English letters. In one operation, you can do the following:

Select any non-empty substring of s, possibly the entire string, then replace each one of its characters with the previous character of the English alphabet. For example, 'b' is converted to 'a', and 'a' is converted to 'z'.

Return the lexicographically smallest string you can obtain after performing the above operation exactly once. A substring is a contiguous sequence of characters in a string. A string x is lexicographically smaller than a string y of the same length if x[i] comes before y[i] in alphabetic order for the first position i such that x[i] != y[i]. Example 1: Input: s = "cbabc" Output: "baabc" Explanation: We apply the operation on the substring starting at index 0, and ending at index 1 inclusive. It can be proven that the resulting string is the lexicographically smallest.

Example 2: Input: s = ”acbbc” Output: ”abaab” Explanation: We apply the operation on the substring starting at index 1, and ending at index 4 inclusive. It can be proven that the resulting string is the lexicographically smallest.

Example 3: Input: s = ”leetcode” Output: ”kddsbncd” Explanation: We apply the operation on the entire string. It can be proven that the resulting string is the lexicographically smallest.

Constraints:

1 <= s.length <= 3 * 105 s consists of lowercase English letters

==
Problem Number: 1626 URL: https://leetcode.com/problems/collecting-chocolates Title: 2735. Collecting Chocolates Problem Description: You are given a 0-indexed integer array nums of size n representing the cost of collecting different chocolates. The cost of collecting the chocolate at the index i is nums[i]. Each chocolate is of a different type, and initially, the chocolate at the index i is of ith type. In one operation, you can do the following with an incurred cost of x:

Simultaneously change the chocolate of ith type to ((i + 1) mod n)th type for all chocolates.

Return the minimum cost to collect chocolates of all types, given that you can perform as many operations as you would like. Example 1: Input: nums = [20,1,15], x = 5 Output: 13 Explanation: Initially, the chocolate types are [0,1,2]. We will buy the 1st type of chocolate at a cost of 1. Now, we will perform the operation at a cost of 5, and the types of chocolates will become [1,2,0]. We will buy the 2nd type of chocolate at a cost of 1. Now, we will again perform the operation at a cost of 5, and the chocolate types will become [2,0,1]. We will buy the 0th type of chocolate at a cost of 1. Thus, the total cost will become (1 + 5 + 1 + 5 + 1) = 13. We can prove that this is optimal.

Example 2: Input: nums = [1,2,3], x = 4 Output: 6 Explanation: We will collect all three types of chocolates at their own price without performing any operations. Therefore, the total cost is 1 + 2 + 3 = 6.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 109 1 <= x <= 109

==
Problem Number: 1627 URL: https://leetcode.com/problems/find-the-value-of-the-partition Title: 2740. Find the Value of the Partition Problem Description: You are given a positive integer array nums. Partition nums into two arrays, nums1 and nums2, such that:

Each element of the array nums belongs to either the array nums1 or the array nums2. Both arrays are non-empty. The value of the partition is minimized.

The value of the partition is |max(nums1) - min(nums2)|. Here, max(nums1) denotes the maximum element of the array nums1, and min(nums2) denotes the minimum element of the array nums2. Return the integer denoting the value of such partition. Example 1: Input: nums = [1,3,2,4] Output: 1 Explanation: We can partition the array nums into nums1 = [1,2] and nums2 = [3,4]. - The maximum element of the array nums1 is equal to 2. - The minimum element of the array nums2 is equal to 3. The value of the partition is |2 - 3| = 1. It can be proven that 1 is the minimum value out of all partitions.

Example 2: Input: nums = [100,1,10] Output: 9 Explanation: We can partition the array nums into nums1 = [10] and nums2 = [100,1]. - The maximum element of the array nums1 is equal to 10. - The minimum element of the array nums2 is equal to 1. The value of the partition is |10 - 1| = 9. It can be proven that 9 is the minimum value out of all partitions.

Constraints:

2 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1628 URL: https://leetcode.com/problems/special-permutations Title: 2741. Special Permutations Problem Description: You are given a 0-indexed integer array nums containing n distinct positive integers. A permutation of nums is called special if:

For all indexes 0 <= i < n - 1, either nums[i] % nums[i+1] == 0 or nums[i+1] % nums[i] == 0.

Return the total number of special permutations. As the answer could be large, return it modulo 109 + 7. Example 1: Input: nums = [2,3,6] Output: 2 Explanation: [3,6,2] and [2,6,3] are the two special permutations of nums.

Example 2: Input: nums = [1,4,3] Output: 2 Explanation: [3,1,4] and [4,1,3] are the two special permutations of nums.

Constraints:

2 <= nums.length <= 14 1 <= nums[i] <= 109

==
Problem Number: 1629 URL: https://leetcode.com/problems/construct-the-longest-new-string Title: 2745. Construct the Longest New String Problem Description: You are given three integers x, y, and z. You have x strings equal to "AA", y strings equal to "BB", and z strings equal to "AB". You want to choose some (possibly all or none) of these strings and concactenate them in some order to form a new string. This new string must not contain "AAA" or "BBB" as a substring. Return the maximum possible length of the new string. A substring is a contiguous non-empty sequence of characters within a string.

Example 1: Input: x = 2, y = 5, z = 1 Output: 12 Explanation: We can concatenate the strings "BB", "AA", "BB", "AA", "BB", and "AB" in that order. Then, our new string is "BBAABBAABBAB". That string has length 12, and we can show that it is impossible to construct a string of longer length.

Example 2: Input: x = 3, y = 2, z = 2 Output: 14 Explanation: We can concatenate the strings "AB", "AB", "AA", "BB", "AA", "BB", and "AA" in that order. Then, our new string is "ABABAABBAABBAA". That string has length 14, and we can show that it is impossible to construct a string of longer length.

Constraints:

1 <= x, y, z <= 50

==
Problem Number: 1630 URL: https://leetcode.com/problems/decremental-string-concatenation Title: 2746. Decremental String Concatenation Problem Description: You are given a 0-indexed array words containing n strings. Let's define a join operation join(x, y) between two strings x and y as concatenating them into xy. However, if the last character of x is equal to the first character of y, one of them is deleted. For example join("ab", "ba") = "aba" and join("ab", "cde") = "abcde". You are to perform n - 1 join operations. Let str0 = words[0]. Starting from i = 1 up to i = n - 1, for the ith operation, you can do one of the following:

Make stri = join(stri - 1, words[i]) Make stri = join(words[i], stri - 1)

Your task is to minimize the length of strn - 1. Return an integer denoting the minimum possible length of strn - 1. Example 1: Input: words = ["aa","ab","bc"] Output: 4 Explanation: In this example, we can perform join operations in the following order to minimize the length of str2: str0 = "aa" str1 = join(str0, "ab") = "aab" str2 = join(str1, "bc") = "aabc" It can be shown that the minimum possible length of str2 is 4. Example 2: Input: words = ["ab","b"] Output: 2 Explanation: In this example, str0 = "ab", there are two ways to get str1: join(str0, "b") = "ab" or join("b", str0) = "bab". The first string, "ab", has the minimum length. Hence, the answer is 2.

Example 3: Input: words = ["aaa","c","aba"] Output: 6 Explanation: In this example, we can perform join operations in the following order to minimize the length of str2: str0 = "aaa" str1 = join(str0, "c") = "aaac" str2 = join("aba", str1) = "abaaac" It can be shown that the minimum possible length of str2 is 6.

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 50 Each character in words[i] is an English lowercase letter

==
Problem Number: 1631 URL: https://leetcode.com/problems/count-zero-

request-servers Title: 2747. Count Zero Request Servers Problem Description: You are given an integer n denoting the total number of servers and a 2D 0-indexed integer array logs, where logs[i] = [server_id, time] denotes that the server with id server_id received a request at time time. You are also given an integer x and a 0-indexed integer array queries. Return a 0-indexed integer array arr of length queries.length where arr[i] represents the number of servers that did not receive any requests during the time interval [queries[i] - x, queries[i]]. Note that the time intervals are inclusive. Example 1: Input: n = 3, logs = [[1,3],[2,6],[1,5]], x = 5, queries = [10,11] Output: [1,2] Explanation: For queries[0]: The servers with ids 1 and 2 get requests in the duration of [5, 10]. Hence, only server 3 gets zero requests. For queries[1]: Only the server with id 2 gets a request in duration of [6,11]. Hence, the servers with ids 1 and 3 are the only servers that do not receive any requests during that time period.

Example 2: Input: n = 3, logs = [[2,4],[2,1],[1,2],[3,1]], x = 2, queries = [3,4] Output: [0,1] Explanation: For queries[0]: All servers get at least one request in the duration of [1, 3]. For queries[1]: Only server with id 3 gets no request in the duration [2,4].

Constraints:

1 <= n <= 105 1 <= logs.length <= 105 1 <= queries.length <= 105 logs[i].length == 2 1 <= logs[i][0] <= n 1 <= logs[i][1] <= 106 1 <= x <= 105 x < queries[i] <= 106

==
Problem Number: 1632 URL: https://leetcode.com/problems/minimum-operations-to-make-the-integer-zero Title: 2749. Minimum Operations to Make the Integer Zero Problem Description: You are given two integers num1 and num2. In one operation, you can choose integer i in the range [0, 60] and subtract 2i + num2 from num1. Return the integer denoting the minimum number of operations needed to make num1 equal to 0. If it is impossible to make num1 equal to 0, return -1. Example 1: Input: num1 = 3, num2 = -2 Output: 3 Explanation: We can make 3 equal to 0 with the following operations: - We choose i = 2 and substract 22 + (-2) from 3, 3 - (4 + (-2)) = 1. - We choose i = 2 and substract 22 + (-2) from 1, 1 - (4 + (-2)) = -1. - We choose i = 0 and substract 20 + (-2) from -1, (-1) - (1 + (-2)) = 0. It can be proven, that 3 is the minimum number of operations that we need to perform.

Example 2: Input: num1 = 5, num2 = 7 Output: -1 Explanation: It can be proven, that it is impossible to make 5 equal to 0 with the given operation.

Constraints:

1 <= num1 <= 109 -109 <= num2 <= 109

==
Problem Number: 1633 URL: https://leetcode.com/problems/ways-to-split-array-into-good-subarrays Title: 2750. Ways to Split Array Into Good Subarrays Problem Description: You are given a binary array nums. A

subarray of an array is good if it contains exactly one element with the value 1. Return an integer denoting the number of ways to split the array nums into good subarrays. As the number may be too large, return it modulo 109 + 7. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [0,1,0,0,1] Output: 3 Explanation: There are 3 ways to split nums into good subarrays: - [0,1] [0,0,1] - [0,1,0] [0,1] - [0,1,0,0] [1]

Example 2: Input: nums = [0,1,0] Output: 1 Explanation: There is 1 way to split nums into good subarrays: - [0,1,0]

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 1

==

Problem Number: 1634 URL: https://leetcode.com/problems/prime-pairs-with-target-sum Title: 2761. Prime Pairs With Target Sum Problem Description: You are given an integer n. We say that two integers x and y form a prime number pair if:

1 <= x <= y <= n x + y == n x and y are prime numbers

Return the 2D sorted list of prime number pairs [xi, yi]. The list should be sorted in increasing order of xi. If there are no prime number pairs at all, return an empty array. Note: A prime number is a natural number greater than 1 with only two factors, itself and 1. Example 1: Input: n = 10 Output: [[3,7],[5,5]] Explanation: In this example, there are two prime pairs that satisfy the criteria. These pairs are [3,7] and [5,5], and we return them in the sorted order as described in the problem statement.

Example 2: Input: n = 2 Output: [] Explanation: We can show that there is no prime number pair that gives a sum of 2, so we return an empty array.

Constraints:

1 <= n <= 106

==

Problem Number: 1635 URL: https://leetcode.com/problems/continuous-subarrays Title: 2762. Continuous Subarrays Problem Description: You are given a 0-indexed integer array nums. A subarray of nums is called continuous if:

Let i, i + 1, ..., j be the indices in the subarray. Then, for each pair of indices i <= i1, i2 <= j, 0 <= |nums[i1] - nums[i2]| <= 2.

Return the total number of continuous subarrays. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [5,4,2,4] Output: 8 Explanation: Continuous subarray of size 1: [5], [4], [2], [4]. Continuous subarray of size 2: [5,4], [4,2], [2,4]. Continuous subarray of size 3: [4,2,4]. Thereare no subarrys of size 4. Total continuous subarrays = 4 + 3 + 1 = 8. It can be shown that there are no more continuous subarrays.

Example 2: Input: nums = [1,2,3] Output: 6 Explanation: Continuous subarray of size 1: [1], [2], [3]. Continuous subarray of size 2: [1,2], [2,3]. Continuous subarray of size 3: [1,2,3]. Total continuous subarrays = 3 + 2 + 1 = 6.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1636 URL: https://leetcode.com/problems/relocate-marbles Title: 2766. Relocate Marbles Problem Description: You are given a 0-indexed integer array nums representing the initial positions of some marbles. You are also given two 0-indexed integer arrays moveFrom and moveTo of equal length. Throughout moveFrom.length steps, you will change the positions of the marbles. On the ith step, you will move all marbles at position moveFrom[i] to position moveTo[i]. After completing all the steps, return the sorted list of occupied positions. Notes:

We call a position occupied if there is at least one marble in that position. There may be multiple marbles in a single position.

Example 1: Input: nums = [1,6,7,8], moveFrom = [1,7,2], moveTo = [2,9,5] Output: [5,6,8,9] Explanation: Initially, the marbles are at positions 1,6,7,8. At the i = 0th step, we move the marbles at position 1 to position 2. Then, positions 2,6,7,8 are occupied. At the i = 1st step, we move the marbles at position 7 to position 9. Then, positions 2,6,8,9 are occupied. At the i = 2nd step, we move the marbles at position 2 to position 5. Then, positions 5,6,8,9 are occupied. At the end, the final positions containing at least one marbles are [5,6,8,9]. Example 2: Input: nums = [1,1,3,3], moveFrom = [1,3], moveTo = [2,2] Output: [2] Explanation: Initially, the marbles are at positions [1,1,3,3]. At the i = 0th step, we move all the marbles at position 1 to position 2. Then, the marbles are at positions [2,2,3,3]. At the i = 1st step, we move all the marbles at position 3 to position 2. Then, the marbles are at positions [2,2,2,2]. Since 2 is the only occupied position, we return [2].

Constraints:

1 <= nums.length <= 105 1 <= moveFrom.length <= 105 moveFrom.length == moveTo.length 1 <= nums[i], moveFrom[i], moveTo[i] <= 109 The test cases are generated such that there is at least a marble in moveFrom[i] at the moment we want to apply the ith move.

==
Problem Number: 1637 URL: https://leetcode.com/problems/partition-string-into-minimum-beautiful-substrings Title: 2767. Partition String Into Minimum Beautiful Substrings Problem Description: Given a binary string s, partition the string into one or more substrings such that each substring is beautiful. A string is beautiful if:

It doesn't contain leading zeros. It's the binary representation of a number that is a power of 5.

Return the minimum number of substrings in such partition. If it is impossible to partition the string s into beautiful substrings, return -1. A substring is a contiguous sequence of characters in a string.   Example 1: Input: s = ”1011” Output: 2 Explanation: We can paritition the given string into [”101”, ”1”]. - The string ”101” does not contain leading zeros and is the binary representation of integer 51 = 5. - The string ”1” does not contain leading zeros and is the binary representation of integer 50 = 1. It can be shown that 2 is the minimum number of beautiful substrings that s can be partitioned into.

Example 2: Input: s = ”111” Output: 3 Explanation: We can paritition the given string into [”1”, ”1”, ”1”]. - The string ”1” does not contain leading zeros and is the binary representation of integer 50 = 1. It can be shown that 3 is the minimum number of beautiful substrings that s can be partitioned into.

Example 3: Input: s = ”0” Output: -1 Explanation: We can not partition the given string into beautiful substrings.

Constraints:

1 <= s.length <= 15 s[i] is either ’0’ or ’1’.

==
Problem Number: 1638 URL: https://leetcode.com/problems/number-of-black-blocks Title: 2768. Number of Black Blocks Problem Description: You are given two integers m and n representing the dimensions of a 0-indexed m x n grid. You are also given a 0-indexed 2D integer matrix coordinates, where coordinates[i] = [x, y] indicates that the cell with coordinates [x, y] is colored black. All cells in the grid that do not appear in coordinates are white. A block is defined as a 2 x 2 submatrix of the grid. More formally, a block with cell [x, y] as its top-left corner where 0 <= x < m - 1 and 0 <= y < n - 1 contains the coordinates [x, y], [x + 1, y], [x, y + 1], and [x + 1, y + 1]. Return a 0-indexed integer array arr of size 5 such that arr[i] is the number of blocks that contains exactly i black cells.   Example 1: Input: m = 3, n = 3, coordinates = [[0,0]] Output: [3,1,0,0,0] Explanation: The grid looks like this:

There is only 1 block with one black cell, and it is the block starting with cell [0,0]. The other 3 blocks start with cells [0,1], [1,0] and [1,1]. They all have zero black cells. Thus, we return [3,1,0,0,0].

Example 2: Input: m = 3, n = 3, coordinates = [[0,0],[1,1],[0,2]] Output: [0,2,2,0,0] Explanation: The grid looks like this:

There are 2 blocks with two black cells (the ones starting with cell coordinates [0,0] and [0,1]). The other 2 blocks have starting cell coordinates of [1,0] and [1,1]. They both have 1 black cell. Therefore, we return [0,2,2,0,0].

Constraints:

2 <= m <= 105 2 <= n <= 105 0 <= coordinates.length <= 104 coordinates[i].length == 2 0 <= coordinates[i][0] < m 0 <= coordinates[i][1] < n It is guaranteed that coordinates contains pairwise distinct coordinates.

==
Problem Number: 1639 URL: https://leetcode.com/problems/maximum-number-of-jumps-to-reach-the-last-index Title: 2770. Maximum Number of Jumps to Reach the Last Index Problem Description: You are given a 0-indexed array nums of n integers and an integer target. You are initially positioned at index 0. In one step, you can jump from index i to any index j such that:

0 <= i < j < n -target <= nums[j] - nums[i] <= target

Return the maximum number of jumps you can make to reach index n - 1. If there is no way to reach index n - 1, return -1. Example 1: Input: nums = [1,3,6,4,1,2], target = 2 Output: 3 Explanation: To go from index 0 to index n - 1 with the maximum number of jumps, you can perform the following jumping sequence: - Jump from index 0 to index 1. - Jump from index 1 to index 3. - Jump from index 3 to index 5. It can be proven that there is no other jumping sequence that goes from 0 to n - 1 with more than 3 jumps. Hence, the answer is 3. Example 2: Input: nums = [1,3,6,4,1,2], target = 3 Output: 5 Explanation: To go from index 0 to index n - 1 with the maximum number of jumps, you can perform the following jumping sequence: - Jump from index 0 to index 1. - Jump from index 1 to index 2. - Jump from index 2 to index 3. - Jump from index 3 to index 4. - Jump from index 4 to index 5. It can be proven that there is no other jumping sequence that goes from 0 to n - 1 with more than 5 jumps. Hence, the answer is 5. Example 3: Input: nums = [1,3,6,4,1,2], target = 0 Output: -1 Explanation: It can be proven that there is no jumping sequence that goes from 0 to n - 1. Hence, the answer is -1.

Constraints:

2 <= nums.length == n <= 1000 -109 <= nums[i] <= 109 0 <= target <= 2 * 109

==
Problem Number: 1640 URL: https://leetcode.com/problems/longest-non-decreasing-subarray-from-two-arrays Title: 2771. Longest Non-decreasing Subarray From Two Arrays Problem Description: You are given two 0-indexed integer arrays nums1 and nums2 of length n. Let's define another 0-indexed integer array, nums3, of length n. For each index i in the range [0, n - 1], you can assign either nums1[i] or nums2[i] to nums3[i]. Your task is to maximize the length of the longest non-decreasing subarray in nums3 by choosing its values optimally. Return an integer representing the length of the longest non-decreasing subarray in nums3. Note: A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums1 = [2,3,1], nums2 = [1,2,1] Output: 2 Explanation: One way to construct nums3 is: nums3 = [nums1[0], nums2[1], nums2[2]] => [2,2,1]. The subarray starting from index 0 and ending at index 1, [2,2], forms a non-decreasing subarray of

length 2. We can show that 2 is the maximum achievable length. Example 2: Input: nums1 = [1,3,2,1], nums2 = [2,2,3,4] Output: 4 Explanation: One way to construct nums3 is: nums3 = [nums1[0], nums2[1], nums2[2], nums2[3]] => [1,2,3,4]. The entire array forms a non-decreasing subarray of length 4, making it the maximum achievable length.

Example 3: Input: nums1 = [1,1], nums2 = [2,2] Output: 2 Explanation: One way to construct nums3 is: nums3 = [nums1[0], nums1[1]] => [1,1]. The entire array forms a non-decreasing subarray of length 2, making it the maximum achievable length.

Constraints:

1 <= nums1.length == nums2.length == n <= 105 1 <= nums1[i], nums2[i] <= 109

==

Problem Number: 1641 URL: https://leetcode.com/problems/apply-operations-to-make-all-array-elements-equal-to-zero Title: 2772. Apply Operations to Make All Array Elements Equal to Zero Problem Description: You are given a 0-indexed integer array nums and a positive integer k. You can apply the following operation on the array any number of times:

Choose any subarray of size k from the array and decrease all its elements by 1.

Return true if you can make all the array elements equal to 0, or false otherwise. A subarray is a contiguous non-empty part of an array. Example 1: Input: nums = [2,2,3,1,1,0], k = 3 Output: true Explanation: We can do the following operations: - Choose the subarray [2,2,3]. The resulting array will be nums = [1,1,2,1,1,0]. - Choose the subarray [2,1,1]. The resulting array will be nums = [1,1,1,0,0,0]. - Choose the subarray [1,1,1]. The resulting array will be nums = [0,0,0,0,0,0].

Example 2: Input: nums = [1,3,1,1], k = 2 Output: false Explanation: It is not possible to make all the array elements equal to 0.

Constraints:

1 <= k <= nums.length <= 105 0 <= nums[i] <= 106

==

Problem Number: 1642 URL: https://leetcode.com/problems/maximum-beauty-of-an-array-after-applying-operation Title: 2779. Maximum Beauty of an Array After Applying Operation Problem Description: You are given a 0-indexed array nums and a non-negative integer k. In one operation, you can do the following:

Choose an index i that hasn't been chosen before from the range [0, nums.length - 1]. Replace nums[i] with any integer from the range [nums[i] - k, nums[i] + k].

The beauty of the array is the length of the longest subsequence consisting of equal elements. Return the maximum possible beauty of the array nums after applying the operation any number of times. Note that you can apply the operation to each index only once. A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the order of the remaining elements.   Example 1: Input: nums = [4,6,1,2], k = 2 Output: 3 Explanation: In this example, we apply the following operations: - Choose index 1, replace it with 4 (from range [4,8]), nums = [4,4,1,2]. - Choose index 3, replace it with 4 (from range [0,4]), nums = [4,4,1,4]. After the applied operations, the beauty of the array nums is 3 (subsequence consisting of indices 0, 1, and 3). It can be proven that 3 is the maximum possible length we can achieve.

Example 2: Input: nums = [1,1,1,1], k = 10 Output: 4 Explanation: In this example we don't have to apply any operations. The beauty of the array nums is 4 (whole array).

Constraints:

1 <= nums.length <= 105 0 <= nums[i], k <= 105

==

Problem Number: 1643 URL: https://leetcode.com/problems/minimum-index-of-a-valid-split Title: 2780. Minimum Index of a Valid Split Problem Description: An element x of an integer array arr of length m is dominant if freq(x) * 2 > m, where freq(x) is the number of occurrences of x in arr. Note that this definition implies that arr can have at most one dominant element. You are given a 0-indexed integer array nums of length n with one dominant element. You can split nums at an index i into two arrays nums[0, ..., i] and nums[i + 1, ..., n - 1], but the split is only valid if:

0 <= i < n - 1 nums[0, ..., i], and nums[i + 1, ..., n - 1] have the same dominant element.

Here, nums[i, ..., j] denotes the subarray of nums starting at index i and ending at index j, both ends being inclusive. Particularly, if j < i then nums[i, ..., j] denotes an empty subarray. Return the minimum index of a valid split. If no valid split exists, return -1.   Example 1: Input: nums = [1,2,2,2] Output: 2 Explanation: We can split the array at index 2 to obtain arrays [1,2,2] and [2]. In array [1,2,2], element 2 is dominant since it occurs twice in the array and 2 * 2 > 3. In array [2], element 2 is dominant since it occurs once in the array and 1 * 2 > 1. Both [1,2,2] and [2] have the same dominant element as nums, so this is a valid split. It can be shown that index 2 is the minimum index of a valid split. Example 2: Input: nums = [2,1,3,1,1,1,7,1,2,1] Output: 4 Explanation: We can split the array at index 4 to obtain arrays [2,1,3,1,1] and [1,7,1,2,1]. In array [2,1,3,1,1], element 1 is dominant since it occurs thrice in the array and 3 * 2 > 5. In array [1,7,1,2,1], element 1 is dominant since it occurs thrice in the array and 3 * 2 > 5. Both [2,1,3,1,1] and [1,7,1,2,1] have the same dominant element as nums, so this is a valid split. It can be shown that index 4 is the minimum index of a valid split. Example 3: Input: nums = [3,3,3,3,7,2,2] Output: -1 Explanation: It can be shown that there is no valid split.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 nums has exactly one dominant element.

==
Problem Number: 1644 URL: https://leetcode.com/problems/sort-vowels-in-a-string Title: 2785. Sort Vowels in a String Problem Description: Given a 0-indexed string s, permute s to get a new string t such that:

All consonants remain in their original places. More formally, if there is an index i with 0 <= i < s.length such that s[i] is a consonant, then t[i] = s[i]. The vowels must be sorted in the nondecreasing order of their ASCII values. More formally, for pairs of indices i, j with 0 <= i < j < s.length such that s[i] and s[j] are vowels, then t[i] must not have a higher ASCII value than t[j].

Return the resulting string. The vowels are 'a', 'e', 'i', 'o', and 'u', and they can appear in lowercase or uppercase. Consonants comprise all letters that are not vowels. Example 1: Input: s = "lEetcOde" Output: "lEOtcede" Explanation: 'E', 'O', and 'e' are the vowels in s; 'l', 't', 'c', and 'd' are all consonants. The vowels are sorted according to their ASCII values, and the consonants remain in the same places.

Example 2: Input: s = "lYmpH" Output: "lYmpH" Explanation: There are no vowels in s (all characters in s are consonants), so we return "lYmpH".

Constraints:

1 <= s.length <= 105 s consists only of letters of the English alphabet in uppercase and lowercase.

==
Problem Number: 1645 URL: https://leetcode.com/problems/visit-array-positions-to-maximize-score Title: 2786. Visit Array Positions to Maximize Score Problem Description: You are given a 0-indexed integer array nums and a positive integer x. You are initially at position 0 in the array and you can visit other positions according to the following rules:

If you are currently in position i, then you can move to any position j such that i < j. For each position i that you visit, you get a score of nums[i]. If you move from a position i to a position j and the parities of nums[i] and nums[j] differ, then you lose a score of x.

Return the maximum total score you can get. Note that initially you have nums[0] points. Example 1: Input: nums = [2,3,6,1,9,2], x = 5 Output: 13 Explanation: We can visit the following positions in the array: 0 -> 2 -> 3 -> 4. The corresponding values are 2, 6, 1 and 9. Since the integers 6 and 1 have different parities, the move 2 -> 3 will make you lose a score of x = 5. The total score will be: 2 + 6 + 1 + 9 - 5 = 13.

Example 2: Input: nums = [2,4,6,8], x = 3 Output: 20 Explanation: All the integers in the array have the same parities, so we can visit all of them without losing any score. The total score is: 2 + 4 + 6 + 8 = 20.

Constraints:

2 <= nums.length <= 105 1 <= nums[i], x <= 106

==

Problem Number: 1646 URL: https://leetcode.com/problems/ways-to-express-an-integer-as-sum-of-powers Title: 2787. Ways to Express an Integer as Sum of Powers Problem Description: Given two positive integers n and x. Return the number of ways n can be expressed as the sum of the xth power of unique positive integers, in other words, the number of sets of unique integers [n1, n2, ..., nk] where n = n1x + n2x + ... + nkx. Since the result can be very large, return it modulo 109 + 7. For example, if n = 160 and x = 3, one way to express n is n = 23 + 33 + 53. Example 1: Input: n = 10, x = 2 Output: 1 Explanation: We can express n as the following: n = 32 + 12 = 10. It can be shown that it is the only way to express 10 as the sum of the 2nd power of unique integers.

Example 2: Input: n = 4, x = 1 Output: 2 Explanation: We can express n in the following ways: - n = 41 = 4. - n = 31 + 11 = 4.

Constraints:

1 <= n <= 300 1 <= x <= 5

==

Problem Number: 1647 URL: https://leetcode.com/problems/largest-element-in-an-array-after-merge-operations Title: 2789. Largest Element in an Array after Merge Operations Problem Description: You are given a 0-indexed array nums consisting of positive integers. You can do the following operation on the array any number of times:

Choose an integer i such that 0 <= i < nums.length - 1 and nums[i] <= nums[i + 1]. Replace the element nums[i + 1] with nums[i] + nums[i + 1] and delete the element nums[i] from the array.

Return the value of the largest element that you can possibly obtain in the final array. Example 1: Input: nums = [2,3,7,9,3] Output: 21 Explanation: We can apply the following operations on the array: - Choose i = 0. The resulting array will be nums = [5,7,9,3]. - Choose i = 1. The resulting array will be nums = [5,16,3]. - Choose i = 0. The resulting array will be nums = [21,3]. The largest element in the final array is 21. It can be shown that we cannot obtain a larger element.

Example 2: Input: nums = [5,3,3] Output: 11 Explanation: We can do the following operations on the array: - Choose i = 1. The resulting array will be nums = [5,6]. - Choose i = 0. The resulting array will be nums = [11]. There is only one element in the final array, which is 11.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 106

==

Problem Number: 1648 URL: https://leetcode.com/problems/count-complete-subarrays-in-an-array Title: 2799. Count Complete Subarrays in an Array Problem Description: You are given an array nums consisting of positive integers. We call a subarray of an array complete if the following condition is satisfied:

The number of distinct elements in the subarray is equal to the number of distinct elements in the whole array.

Return the number of complete subarrays. A subarray is a contiguous non-empty part of an array.   Example 1: Input: nums = [1,3,1,2,2] Output: 4 Explanation: The complete subarrays are the following: [1,3,1,2], [1,3,1,2,2], [3,1,2] and [3,1,2,2].

Example 2: Input: nums = [5,5,5,5] Output: 10 Explanation: The array consists only of the integer 5, so any subarray is complete. The number of subarrays that we can choose is 10.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 2000

==

Problem Number: 1649 URL: https://leetcode.com/problems/shortest-string-that-contains-three-strings Title: 2800. Shortest String That Contains Three Strings Problem Description: Given three strings a, b, and c, your task is to find a string that has the minimum length and contains all three strings as substrings. If there are multiple such strings, return the lexicographically smallest one. Return a string denoting the answer to the problem. Notes

A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b. A substring is a contiguous sequence of characters within a string.

Example 1: Input: a = "abc", b = "bca", c = "aaa" Output: "aaabca" Explanation: We show that "aaabca" contains all the given strings: a = ans[2...4], b = ans[3..5], c = ans[0..2]. It can be shown that the length of the resulting string would be at least 6 and "aaabca" is the lexicographically smallest one. Example 2: Input: a = "ab", b = "ba", c = "aba" Output: "aba" Explanation: We show that the string "aba" contains all the given strings: a = ans[0..1], b = ans[1..2], c = ans[0..2]. Since the length of c is 3, the length of the resulting string would be at least 3. It can be shown that "aba" is the lexicographically smallest one.

Constraints:

1 <= a.length, b.length, c.length <= 100 a, b, c consist only of lowercase English letters.

== Problem Number: 1650 URL: https://leetcode.com/problems/insert-greatest-common-divisors-in-linked-list Title: 2807. Insert Greatest Common Divisors in Linked List Problem Description: Given the head of a linked list head, in which each node contains an integer value. Between every pair of adjacent nodes, insert a new node with a value equal to the greatest common divisor of them. Return the linked list after insertion. The greatest common divisor of two numbers is the largest positive integer that evenly divides both numbers. Example 1:

Input: head = [18,6,10,3] Output: [18,6,6,2,10,1,3] Explanation: The 1st diagram denotes the initial linked list and the 2nd diagram denotes the linked list after inserting the new nodes (nodes in blue are the inserted nodes). - We insert the greatest common divisor of 18 and 6 = 6 between the 1st and the 2nd nodes. - We insert the greatest common divisor of 6 and 10 = 2 between the 2nd and the 3rd nodes. - We insert the greatest common divisor of 10 and 3 = 1 between the 3rd and the 4th nodes. There are no more adjacent nodes, so we return the linked list.

Example 2:

Input: head = [7] Output: [7] Explanation: The 1st diagram denotes the initial linked list and the 2nd diagram denotes the linked list after inserting the new nodes. There are no pairs of adjacent nodes, so we return the initial linked list.

Constraints:

The number of nodes in the list is in the range [1, 5000]. 1 <= Node.val <= 1000

== Problem Number: 1651 URL: https://leetcode.com/problems/minimum-seconds-to-equalize-a-circular-array Title: 2808. Minimum Seconds to Equalize a Circular Array Problem Description: You are given a 0-indexed array nums containing n integers. At each second, you perform the following operation on the array:

For every index i in the range [0, n - 1], replace nums[i] with either nums[i], nums[(i - 1 + n) % n], or nums[(i + 1) % n].

Note that all the elements get replaced simultaneously. Return the minimum number of seconds needed to make all elements in the array nums equal. Example 1: Input: nums = [1,2,1,2] Output: 1 Explanation: We can equalize the array in 1 second in the following way: - At 1st second, replace values at each index with [nums[3],nums[1],nums[3],nums[3]]. After replacement, nums = [2,2,2,2]. It can be proven that 1 second is the minimum amount of seconds needed for equalizing the array.

Example 2: Input: nums = [2,1,3,3,2] Output: 2 Explanation: We can equalize the array in 2 seconds in the following way: - At 1st second, replace values at each index with [nums[0],nums[2],nums[2],nums[2],nums[3]]. After replacement, nums = [2,3,3,3,3]. - At 2nd second, replace values at each index with [nums[1],nums[1],nums[2],nums[3],nums[4]]. After replacement, nums = [3,3,3,3,3]. It can be proven that 2 seconds is the minimum amount of seconds needed for equalizing the array.

Example 3: Input: nums = [5,5,5,5] Output: 0 Explanation: We don't need to perform any operations as all elements in the initial array are the same.

Constraints:

1 <= n == nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 1652 URL: https://leetcode.com/problems/check-if-it-is-possible-to-split-array Title: 2811. Check if it is Possible to Split Array Problem Description: You are given an array nums of length n and an integer m. You need to determine if it is possible to split the array into n non-empty arrays by performing a series of steps. In each step, you can select an existing array (which may be the result of previous steps) with a length of at least two and split it into two subarrays, if, for each resulting subarray, at least one of the following holds:

The length of the subarray is one, or The sum of elements of the subarray is greater than or equal to m.

Return true if you can split the given array into n arrays, otherwise return false. Note: A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [2, 2, 1], m = 4 Output: true Explanation: We can split the array into [2, 2] and [1] in the first step. Then, in the second step, we can split [2, 2] into [2] and [2]. As a result, the answer is true. Example 2: Input: nums = [2, 1, 3], m = 5 Output: false Explanation: We can try splitting the array in two different ways: the first way is to have [2, 1] and [3], and the second way is to have [2] and [1, 3]. However, both of these ways are not valid. So, the answer is false. Example 3: Input: nums = [2, 3, 3, 2, 3], m = 6 Output: true Explanation: We can split the array into [2, 3, 3, 2] and [3] in the first step. Then, in the second step, we can split [2, 3, 3, 2] into [2, 3, 3] and [2]. Then, in the third step, we can split [2, 3, 3] into [2] and [3, 3]. And in the last step we can split [3, 3] into [3] and [3]. As a result, the answer is true.

Constraints:

1 <= n == nums.length <= 100 1 <= nums[i] <= 100 1 <= m <= 200

==
Problem Number: 1653 URL: https://leetcode.com/problems/find-the-safest-path-in-a-grid Title: 2812. Find the Safest Path in a Grid Problem Description: You are given a 0-indexed 2D matrix grid of size n x n, where (r, c) represents:

A cell containing a thief if grid[r][c] = 1 An empty cell if grid[r][c] = 0

You are initially positioned at cell (0, 0). In one move, you can move to any adjacent cell in the grid, including cells containing thieves. The safeness factor of a path on the grid is defined as the minimum manhattan distance from any cell in the path to any thief in the grid. Return the maximum safeness factor of all paths leading to cell (n - 1, n - 1). An adjacent cell of cell (r, c), is one of the cells (r, c + 1), (r, c - 1), (r + 1, c) and (r - 1, c) if it exists. The Manhattan distance between two cells (a, b) and (x, y) is equal to |a - x| + |b - y|, where |val| denotes the absolute value of val.   Example 1:

Input: grid = [[1,0,0],[0,0,0],[0,0,1]] Output: 0 Explanation: All paths from (0, 0) to (n - 1, n - 1) go through the thieves in cells (0, 0) and (n - 1, n - 1).

Example 2:

Input: grid = [[0,0,1],[0,0,0],[0,0,0]] Output: 2 Explanation: The path depicted in the picture above has a safeness factor of 2 since: - The closest cell of the path to the thief at cell (0, 2) is cell (0, 0). The distance between them is | 0 - 0 | + | 0 - 2 | = 2. It can be shown that there are no other paths with a higher safeness factor.

Example 3:

Input: grid = [[0,0,0,1],[0,0,0,0],[0,0,0,0],[1,0,0,0]] Output: 2 Explanation: The path depicted in the picture above has a safeness factor of 2 since: - The closest cell of the path to the thief at cell (0, 3) is cell (1, 2). The distance between them is | 0 - 1 | + | 3 - 2 | = 2. - The closest cell of the path to the thief at cell (3, 0) is cell (3, 2). The distance between them is | 3 - 3 | + | 0 - 2 | = 2. It can be shown that there are no other paths with a higher safeness factor.

Constraints:

1 <= grid.length == n <= 400 grid[i].length == n grid[i][j] is either 0 or 1. There is at least one thief in the grid.

==
Problem Number: 1654 URL: https://leetcode.com/problems/double-a-number-represented-as-a-linked-list Title: 2816. Double a Number Represented as a Linked List Problem Description: You are given the head of a non-empty linked list representing a non-negative integer without leading zeroes. Return the head of the linked list after doubling it.   Example 1:

Input: head = [1,8,9] Output: [3,7,8] Explanation: The figure above corresponds to the given linked list which represents the number 189. Hence, the returned linked list represents the number 189 * 2 = 378.

Example 2:

Input: head = [9,9,9] Output: [1,9,9,8] Explanation: The figure above corresponds to the given linked list which represents the number 999. Hence, the returned linked list reprersents the number 999 * 2 = 1998.

Constraints:

The number of nodes in the list is in the range [1, 104] 0 <= Node.val <= 9 The input is generated such that the list represents a number that does not have leading zeros, except the number 0 itself.

==
Problem Number: 1655 URL: https://leetcode.com/problems/minimum-absolute-difference-between-elements-with-constraint Title: 2817. Minimum Absolute Difference Between Elements With Constraint Problem Description: You are given a 0-indexed integer array nums and an integer x. Find the minimum absolute difference between two elements in the array that are at least x indices apart. In other words, find two indices i and j such that abs(i - j) >= x and abs(nums[i] - nums[j]) is minimized. Return an integer denoting the minimum absolute difference between two elements that are at least x indices apart. Example 1: Input: nums = [4,3,2,4], x = 2 Output: 0 Explanation: We can select nums[0] = 4 and nums[3] = 4. They are at least 2 indices apart, and their absolute difference is the minimum, 0. It can be shown that 0 is the optimal answer.

Example 2: Input: nums = [5,3,2,10,15], x = 1 Output: 1 Explanation: We can select nums[1] = 3 and nums[2] = 2. They are at least 1 index apart, and their absolute difference is the minimum, 1. It can be shown that 1 is the optimal answer.

Example 3: Input: nums = [1,2,3,4], x = 3 Output: 3 Explanation: We can select nums[0] = 1 and nums[3] = 4. They are at least 3 indices apart, and their absolute difference is the minimum, 3. It can be shown that 3 is the optimal answer.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 0 <= x < nums.length

==
Problem Number: 1656 URL: https://leetcode.com/problems/make-string-a-subsequence-using-cyclic-increments Title: 2825. Make String a Subsequence Using Cyclic Increments Problem Description: You are given two 0-indexed strings str1 and str2. In an operation, you select a set of indices in str1, and for each index i in the set, increment str1[i] to the next character cyclically. That is 'a' becomes 'b', 'b' becomes 'c', and so on, and 'z' becomes 'a'. Return true if it is possible to make str2 a subsequence of str1 by performing the operation at most once, and false otherwise. Note: A subsequence of a string is a new string that is formed from the original string by deleting some (possibly none) of the characters without disturbing the relative positions of the remaining characters. Example 1: Input: str1 = "abc", str2 = "ad" Output: true Explanation: Select index 2 in str1. Increment str1[2] to become 'd'. Hence, str1 becomes "abd" and str2 is now a subsequence. Therefore, true is returned. Example 2: Input: str1 = "zc", str2 = "ad" Output: true Explanation: Select

indices 0 and 1 in str1. Increment str1[0] to become 'a'. Increment str1[1] to become 'd'. Hence, str1 becomes "ad" and str2 is now a subsequence. Therefore, true is returned. Example 3: Input: str1 = "ab", str2 = "d" Output: false Explanation: In this example, it can be shown that it is impossible to make str2 a subsequence of str1 using the operation at most once. Therefore, false is returned. Constraints:

1 <= str1.length <= 105 1 <= str2.length <= 105 str1 and str2 consist of only lowercase English letters.

==
Problem Number: 1657 URL: https://leetcode.com/problems/sorting-three-groups Title: 2826. Sorting Three Groups Problem Description: You are given a 0-indexed integer array nums of length n.

The numbers from 0 to n - 1 are divided into three groups numbered from 1 to 3, where number i belongs to group nums[i]. Notice that some groups may be empty.

You are allowed to perform this operation any number of times:

Pick number x and change its group. More formally, change nums[x] to any number from 1 to 3.

A new array res is constructed using the following procedure:

Sort the numbers in each group independently. Append the elements of groups 1, 2, and 3 to res in this order.

Array nums is called a beautiful array if the constructed array res is sorted in non-decreasing order. Return the minimum number of operations to make nums a beautiful array. Example 1: Input: nums = [2,1,3,2,1] Output: 3 Explanation: It's optimal to perform three operations: 1. change nums[0] to 1. 2. change nums[2] to 1. 3. change nums[3] to 1. After performing the operations and sorting the numbers in each group, group 1 becomes equal to [0,1,2,3,4] and group 2 and group 3 become empty. Hence, res is equal to [0,1,2,3,4] which is sorted in non-decreasing order. It can be proven that there is no valid sequence of less than three operations.

Example 2: Input: nums = [1,3,2,1,3,3] Output: 2 Explanation: It's optimal to perform two operations: 1. change nums[1] to 1. 2. change nums[2] to 1. After performing the operations and sorting the numbers in each group, group 1 becomes equal to [0,1,2,3], group 2 becomes empty, and group 3 becomes equal to [4,5]. Hence, res is equal to [0,1,2,3,4,5] which is sorted in non-decreasing order. It can be proven that there is no valid sequence of less than two operations.

Example 3: Input: nums = [2,2,2,2,3,3] Output: 0 Explanation: It's optimal to not perform operations. After sorting the numbers in each group, group 1 becomes empty, group 2 becomes equal to [0,1,2,3] and group 3 becomes equal to [4,5]. Hence, res is equal to [0,1,2,3,4,5] which is sorted in non-decreasing order.

Constraints:

1 <= nums.length <= 100 1 <= nums[i] <= 3

==
Problem Number: 1658 URL: https://leetcode.com/problems/determine-the-minimum-sum-of-a-k-avoiding-array Title: 2829. Determine the Minimum Sum of a k-avoiding Array Problem Description: You are given two integers, n and k. An array of distinct positive integers is called a k-avoiding array if there does not exist any pair of distinct elements that sum to k. Return the minimum possible sum of a k-avoiding array of length n.   Example 1: Input: n = 5, k = 4 Output: 18 Explanation: Consider the k-avoiding array [1,2,4,5,6], which has a sum of 18. It can be proven that there is no k-avoiding array with a sum less than 18.

Example 2: Input: n = 2, k = 6 Output: 3 Explanation: We can construct the array [1,2], which has a sum of 3. It can be proven that there is no k-avoiding array with a sum less than 3.

Constraints:

1 <= n, k <= 50

==
Problem Number: 1659 URL: https://leetcode.com/problems/maximize-the-profit-as-the-salesman Title: 2830. Maximize the Profit as the Salesman Problem Description: You are given an integer n representing the number of houses on a number line, numbered from 0 to n - 1. Additionally, you are given a 2D integer array offers where offers[i] = [starti, endi, goldi], indicating that ith buyer wants to buy all the houses from starti to endi for goldi amount of gold. As a salesman, your goal is to maximize your earnings by strategically selecting and selling houses to buyers. Return the maximum amount of gold you can earn. Note that different buyers can't buy the same house, and some houses may remain unsold.   Example 1: Input: n = 5, offers = [[0,0,1],[0,2,2],[1,3,2]] Output: 3 Explanation: There are 5 houses numbered from 0 to 4 and there are 3 purchase offers. We sell houses in the range [0,0] to 1st buyer for 1 gold and houses in the range [1,3] to 3rd buyer for 2 golds. It can be proven that 3 is the maximum amount of gold we can achieve.

Example 2: Input: n = 5, offers = [[0,0,1],[0,2,10],[1,3,2]] Output: 10 Explanation: There are 5 houses numbered from 0 to 4 and there are 3 purchase offers. We sell houses in the range [0,2] to 2nd buyer for 10 golds. It can be proven that 10 is the maximum amount of gold we can achieve.

Constraints:

1 <= n <= 105 1 <= offers.length <= 105 offers[i].length == 3 0 <= starti <= endi <= n - 1 1 <= goldi <= 103

==
Problem Number: 1660 URL: https://leetcode.com/problems/find-the-longest-

equal-subarray Title: 2831. Find the Longest Equal Subarray Problem Description: You are given a 0-indexed integer array nums and an integer k. A subarray is called equal if all of its elements are equal. Note that the empty subarray is an equal subarray. Return the length of the longest possible equal subarray after deleting at most k elements from nums. A subarray is a contiguous, possibly empty sequence of elements within an array. Example 1: Input: nums = [1,3,2,3,1,3], k = 3 Output: 3 Explanation: It's optimal to delete the elements at index 2 and index 4. After deleting them, nums becomes equal to [1, 3, 3, 3]. The longest equal subarray starts at i = 1 and ends at j = 3 with length equal to 3. It can be proven that no longer equal subarrays can be created.

Example 2: Input: nums = [1,1,2,2,1,1], k = 2 Output: 4 Explanation: It's optimal to delete the elements at index 2 and index 3. After deleting them, nums becomes equal to [1, 1, 1, 1]. The array itself is an equal subarray, so the answer is 4. It can be proven that no longer equal subarrays can be created.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= nums.length 0 <= k <= nums.length

==
Problem Number: 1661 URL: https://leetcode.com/problems/find-the-minimum-possible-sum-of-a-beautiful-array Title: 2834. Find the Minimum Possible Sum of a Beautiful Array Problem Description: You are given positive integers n and target. An array nums is beautiful if it meets the following conditions:

nums.length == n. nums consists of pairwise distinct positive integers. There doesn't exist two distinct indices, i and j, in the range [0, n - 1], such that nums[i] + nums[j] == target.

Return the minimum possible sum that a beautiful array could have modulo 109 + 7. Example 1: Input: n = 2, target = 3 Output: 4 Explanation: We can see that nums = [1,3] is beautiful. - The array nums has length n = 2. - The array nums consists of pairwise distinct positive integers. - There doesn't exist two distinct indices, i and j, with nums[i] + nums[j] == 3. It can be proven that 4 is the minimum possible sum that a beautiful array could have.

Example 2: Input: n = 3, target = 3 Output: 8 Explanation: We can see that nums = [1,3,4] is beautiful. - The array nums has length n = 3. - The array nums consists of pairwise distinct positive integers. - There doesn't exist two distinct indices, i and j, with nums[i] + nums[j] == 3. It can be proven that 8 is the minimum possible sum that a beautiful array could have.

Example 3: Input: n = 1, target = 1 Output: 1 Explanation: We can see, that nums = [1] is beautiful.

Constraints:

1 <= n <= 109 1 <= target <= 109

==

Problem Number: 1662 URL: https://leetcode.com/problems/check-if-strings-can-be-made-equal-with-operations-ii Title: 2840. Check if Strings Can be Made Equal With Operations II Problem Description: You are given two strings s1 and s2, both of length n, consisting of lowercase English letters. You can apply the following operation on any of the two strings any number of times:

Choose any two indices i and j such that i < j and the difference j - i is even, then swap the two characters at those indices in the string.

Return true if you can make the strings s1 and s2 equal, and false otherwise. Example 1: Input: s1 = "abcdba", s2 = "cabdab" Output: true Explanation: We can apply the following operations on s1: - Choose the indices i = 0, j = 2. The resulting string is s1 = "cbadba". - Choose the indices i = 2, j = 4. The resulting string is s1 = "cbbdaa". - Choose the indices i = 1, j = 5. The resulting string is s1 = "cabdab" = s2.

Example 2: Input: s1 = "abe", s2 = "bea" Output: false Explanation: It is not possible to make the two strings equal.

Constraints:

n == s1.length == s2.length 1 <= n <= 105 s1 and s2 consist only of lowercase English letters.

==

Problem Number: 1663 URL: https://leetcode.com/problems/maximum-sum-of-almost-unique-subarray Title: 2841. Maximum Sum of Almost Unique Subarray Problem Description: You are given an integer array nums and two positive integers m and k. Return the maximum sum out of all almost unique subarrays of length k of nums. If no such subarray exists, return 0. A subarray of nums is almost unique if it contains at least m distinct elements. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [2,6,7,3,1,7], m = 3, k = 4 Output: 18 Explanation: There are 3 almost unique subarrays of size k = 4. These subarrays are [2, 6, 7, 3], [6, 7, 3, 1], and [7, 3, 1, 7]. Among these subarrays, the one with the maximum sum is [2, 6, 7, 3] which has a sum of 18.

Example 2: Input: nums = [5,9,9,2,4,5,4], m = 1, k = 3 Output: 23 Explanation: There are 5 almost unique subarrays of size k. These subarrays are [5, 9, 9], [9, 9, 2], [9, 2, 4], [2, 4, 5], and [4, 5, 4]. Among these subarrays, the one with the maximum sum is [5, 9, 9] which has a sum of 23.

Example 3: Input: nums = [1,2,1,2,1,2,1], m = 3, k = 3 Output: 0 Explanation: There are no subarrays of size k = 3 that contain at least m = 3 distinct elements in the given array [1,2,1,2,1,2,1]. Therefore, no almost unique subarrays exist, and the maximum sum is 0.

Constraints:

1 <= nums.length <= 2 * 104 1 <= m <= k <= nums.length 1 <= nums[i] <= 109

==
Problem Number: 1664 URL: https://leetcode.com/problems/minimum-operations-to-make-a-special-number Title: 2844. Minimum Operations to Make a Special Number Problem Description: You are given a 0-indexed string num representing a non-negative integer. In one operation, you can pick any digit of num and delete it. Note that if you delete all the digits of num, num becomes 0. Return the minimum number of operations required to make num special. An integer x is considered special if it is divisible by 25. Example 1: Input: num = "2245047" Output: 2 Explanation: Delete digits num[5] and num[6]. The resulting number is "22450" which is special since it is divisible by 25. It can be shown that 2 is the minimum number of operations required to get a special number. Example 2: Input: num = "2908305" Output: 3 Explanation: Delete digits num[3], num[4], and num[6]. The resulting number is "2900" which is special since it is divisible by 25. It can be shown that 3 is the minimum number of operations required to get a special number. Example 3: Input: num = "10" Output: 1 Explanation: Delete digit num[0]. The resulting number is "0" which is special since it is divisible by 25. It can be shown that 1 is the minimum number of operations required to get a special number.

Constraints:

1 <= num.length <= 100 num only consists of digits '0' through '9'. num does not contain any leading zeros.

==
Problem Number: 1665 URL: https://leetcode.com/problems/count-of-interesting-subarrays Title: 2845. Count of Interesting Subarrays Problem Description: You are given a 0-indexed integer array nums, an integer modulo, and an integer k. Your task is to find the count of subarrays that are interesting. A subarray nums[l..r] is interesting if the following condition holds:

Let cnt be the number of indices i in the range [l, r] such that nums[i] % modulo == k. Then, cnt % modulo == k.

Return an integer denoting the count of interesting subarrays. Note: A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [3,2,4], modulo = 2, k = 1 Output: 3 Explanation: In this example the interesting subarrays are: The subarray nums[0..0] which is [3]. - There is only one index, i = 0, in the range [0, 0] that satisfies nums[i] % modulo == k. - Hence, cnt = 1 and cnt % modulo == k. The subarray nums[0..1] which is [3,2]. - There is only one index, i = 0, in the range [0, 1] that satisfies nums[i] % modulo == k. - Hence, cnt = 1 and cnt % modulo == k. The subarray nums[0..2] which is [3,2,4]. - There is only one index, i = 0, in the range [0, 2] that satisfies nums[i] % modulo == k. - Hence, cnt = 1 and cnt % modulo == k. It can be shown that there are no other interesting subarrays. So, the answer is 3. Example 2: Input: nums = [3,1,9,6], modulo = 3, k = 0 Output: 2 Explanation: In this example the interesting subarrays are: The subarray nums[0..3] which is [3,1,9,6]. - There are three indices, i = 0, 2, 3, in the range [0, 3] that satisfy nums[i] % modulo == k. - Hence, cnt = 3 and cnt % modulo == k. The subarray nums[1..1] which is [1]. - There is no index, i, in the range [1, 1] that satisfies nums[i] % modulo == k. - Hence, cnt = 0 and cnt % modulo == k. It can be shown that there are no other interesting subarrays. So, the answer is 2.   Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= modulo <= 109 0 <= k < modulo

==
Problem Number: 1666 URL: https://leetcode.com/problems/determine-if-a-cell-is-reachable-at-a-given-time Title: 2849. Determine if a Cell Is Reachable at a Given Time Problem Description: You are given four integers sx, sy, fx, fy, and a non-negative integer t. In an infinite 2D grid, you start at the cell (sx, sy). Each second, you must move to any of its adjacent cells. Return true if you can reach cell (fx, fy) after exactly t seconds, or false otherwise. A cell's adjacent cells are the 8 cells around it that share at least one corner with it. You can visit the same cell several times.   Example 1:

Input: sx = 2, sy = 4, fx = 7, fy = 7, t = 6 Output: true Explanation: Starting at cell (2, 4), we can reach cell (7, 7) in exactly 6 seconds by going through the cells depicted in the picture above.

Example 2:

Input: sx = 3, sy = 1, fx = 7, fy = 3, t = 3 Output: false Explanation: Starting at cell (3, 1), it takes at least 4 seconds to reach cell (7, 3) by going through the cells depicted in the picture above. Hence, we cannot reach cell (7, 3) at the third second.

Constraints:

1 <= sx, sy, fx, fy <= 109 0 <= t <= 109

==
Problem Number: 1667 URL: https://leetcode.com/problems/minimum-moves-to-spread-stones-over-grid Title: 2850. Minimum Moves to Spread Stones Over Grid Problem Description: You are given a 0-indexed 2D integer matrix grid of size 3 * 3, representing the number of stones in each cell. The grid contains exactly 9 stones, and there can be multiple stones in a single cell. In one move, you can move a single stone from its current cell to any other cell if the two cells share a side. Return the minimum number of moves required to place one stone in each cell.   Example 1:

Input: grid = [[1,1,0],[1,1,1],[1,2,1]] Output: 3 Explanation: One possible sequence of moves to place one stone in each cell is: 1- Move one stone from cell (2,1) to cell (2,2). 2- Move one stone from cell (2,2) to cell (1,2). 3- Move one stone from cell (1,2) to cell (0,2). In total, it takes 3 moves to place one stone in each cell of the grid. It can be shown that 3 is the minimum number of moves required to place one stone in each cell.

Example 2:

Input: grid = [[1,3,0],[1,0,0],[1,0,3]] Output: 4 Explanation: One possible sequence of moves to place one stone in each cell is: 1- Move one stone from cell (0,1) to cell (0,2). 2- Move one stone from cell (0,1) to cell (1,1). 3- Move one stone from cell (2,2) to cell (1,2). 4- Move one stone from cell (2,2) to cell (2,1). In total, it takes 4 moves to place one stone in each cell of the grid. It can be shown that 4 is the minimum number of moves required to place one stone in each cell.

Constraints:

grid.length == grid[i].length == 3 0 <= grid[i][j] <= 9 Sum of grid is equal to 9.

==
Problem Number: 1668 URL: https://leetcode.com/problems/minimum-array-length-after-pair-removals Title: 2856. Minimum Array Length After Pair Removals Problem Description: You are given a 0-indexed sorted array of integers nums. You can perform the following operation any number of times:

Choose two indices, i and j, where i < j, such that nums[i] < nums[j]. Then, remove the elements at indices i and j from nums. The remaining elements retain their original order, and the array is re-indexed.

Return an integer that denotes the minimum length of nums after performing the operation any number of times (including zero). Note that nums is sorted in non-decreasing order. Example 1: Input: nums = [1,3,4,9] Output: 0 Explanation: Initially, nums = [1, 3, 4, 9]. In the first operation, we can choose index 0 and 1 because nums[0] < nums[1] <=> 1 < 3. Remove indices 0 and 1, and nums becomes [4, 9]. For the next operation, we can choose index 0 and 1 because nums[0] < nums[1] <=> 4 < 9. Remove indices 0 and 1, and nums becomes an empty array []. Hence, the minimum length achievable is 0. Example 2: Input: nums = [2,3,6,9] Output: 0 Explanation: Initially, nums = [2, 3, 6, 9]. In the first operation, we can choose index 0 and 2 because nums[0] < nums[2] <=> 2 < 6. Remove indices 0 and 2, and nums becomes [3, 9]. For the next operation, we can choose index 0 and 1 because nums[0] < nums[1] <=> 3 < 9. Remove indices 0 and 1, and nums becomes an empty array []. Hence, the minimum length achievable is 0.

Example 3: Input: nums = [1,1,2] Output: 1 Explanation: Initially, nums = [1, 1, 2]. In an operation, we can choose index 0 and 2 because nums[0] < nums[2] <=> 1 < 2. Remove indices 0 and 2, and nums becomes [1]. It is no longer possible to perform an operation on the array. Hence, the minimum achievable length is 1.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 nums is sorted in non-decreasing order.

==

Problem Number: 1669 URL: https://leetcode.com/problems/count-pairs-of-points-with-distance-k Title: 2857. Count Pairs of Points With Distance k Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1: Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2: Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1670 URL: https://leetcode.com/problems/happy-students Title: 2860. Happy Students Problem Description: You are given a 0-indexed integer array nums of length n where n is the total number of students in the class. The class teacher tries to select a group of students so that all the students remain happy. The ith student will become happy if one of these two conditions is met:

The student is selected and the total number of selected students is strictly greater than nums[i]. The student is not selected and the total number of selected students is strictly less than nums[i].

Return the number of ways to select a group of students so that everyone remains happy.   Example 1: Input: nums = [1,1] Output: 2 Explanation: The two possible ways are: The class teacher selects no student. The class teacher selects both students to form the group. If the class teacher selects just one student to form a group then the both students will not be happy. Therefore, there are only two possible ways.

Example 2: Input: nums = [6,0,3,3,6,7,2,7] Output: 3 Explanation: The three possible ways are: The class teacher selects the student with index = 1 to form the group. The class teacher selects the students with index = 1, 2, 3, 6 to form the group. The class teacher selects all the students to form the group.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] < nums.length

==

Problem Number: 1671 URL: https://leetcode.com/problems/maximum-number-of-alloys Title: 2861. Maximum Number of Alloys Problem Description: You are the owner of a company that creates alloys using various types of metals. There are n different types of metals available, and you have access to k machines that can be used to create alloys. Each machine requires a specific amount of each metal type to create an alloy. For the ith machine to create an alloy, it needs composition[i][j] units of metal of type j. Initially, you have stock[i] units of metal type i, and purchasing one unit of metal type i costs cost[i] coins. Given integers n, k, budget, a 1-indexed 2D array composition, and 1-indexed arrays stock and cost, your goal is to maximize the number of alloys the company can create while staying within the budget of budget coins. All alloys must be created with the same machine. Return the maximum number of alloys that the company can create. Example 1: Input: n = 3, k = 2, budget = 15, composition = [[1,1,1],[1,1,10]], stock = [0,0,0], cost = [1,2,3] Output: 2 Explanation: It is optimal to use the 1st machine to create alloys. To create 2 alloys we need to buy the: - 2 units of metal of the 1st type. - 2 units of metal of the 2nd type. - 2 units of metal of the 3rd type. In total, we need 2 * 1 + 2 * 2 + 2 * 3 = 12 coins, which is smaller than or equal to budget = 15. Notice that we have 0 units of metal of each type and we have to buy all the required units of metal. It can be proven that we can create at most 2 alloys.

Example 2: Input: n = 3, k = 2, budget = 15, composition = [[1,1,1],[1,1,10]], stock = [0,0,100], cost = [1,2,3] Output: 5 Explanation: It is optimal to use the 2nd machine to create alloys. To create 5 alloys we need to buy: - 5 units of metal of the 1st type. - 5 units of metal of the 2nd type. - 0 units of metal of the 3rd type. In total, we need 5 * 1 + 5 * 2 + 0 * 3 = 15 coins, which is smaller than or equal to budget = 15. It can be proven that we can create at most 5 alloys.

Example 3: Input: n = 2, k = 3, budget = 10, composition = [[2,1],[1,2],[1,1]], stock = [1,1], cost = [5,5] Output: 2 Explanation: It is optimal to use the 3rd machine to create alloys. To create 2 alloys we need to buy the: - 1 unit of metal of the 1st type. - 1 unit of metal of the 2nd type. In total, we need 1 * 5 + 1 * 5 = 10 coins, which is smaller than or equal to budget = 10. It can be proven that we can create at most 2 alloys.

Constraints:

1 <= n, k <= 100 0 <= budget <= 108 composition.length == k composition[i].length == n 1 <= composition[i][j] <= 100 stock.length == cost.length

== n 0 <= stock[i] <= 108 1 <= cost[i] <= 100

==
Problem Number: 1672 URL: https://leetcode.com/problems/beautiful-towers-i Title: 2865. Beautiful Towers I Problem Description: You are given a 0-indexed array maxHeights of n integers. You are tasked with building n towers in the coordinate line. The ith tower is built at coordinate i and has a height of heights[i]. A configuration of towers is beautiful if the following conditions hold:

1 <= heights[i] <= maxHeights[i] heights is a mountain array.

Array heights is a mountain if there exists an index i such that:

For all 0 < j <= i, heights[j - 1] <= heights[j] For all i <= k < n - 1, heights[k + 1] <= heights[k]

Return the maximum possible sum of heights of a beautiful configuration of towers. Example 1: Input: maxHeights = [5,3,4,1,1] Output: 13 Explanation: One beautiful configuration with a maximum sum is heights = [5,3,3,1,1]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 0. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 13. Example 2: Input: maxHeights = [6,5,3,9,2,7] Output: 22 Explanation: One beautiful configuration with a maximum sum is heights = [3,3,3,9,2,2]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 3. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 22. Example 3: Input: maxHeights = [3,2,5,5,2,3] Output: 18 Explanation: One beautiful configuration with a maximum sum is heights = [2,2,5,5,2,2]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 2. Note that, for this configuration, i = 3 can also be considered a peak. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 18.

Constraints:

1 <= n == maxHeights <= 103 1 <= maxHeights[i] <= 109

==
Problem Number: 1673 URL: https://leetcode.com/problems/beautiful-towers-ii Title: 2866. Beautiful Towers II Problem Description: You are given a 0-indexed array maxHeights of n integers. You are tasked with building n towers in the coordinate line. The ith tower is built at coordinate i and has a height of heights[i]. A configuration of towers is beautiful if the following conditions hold:

1 <= heights[i] <= maxHeights[i] heights is a mountain array.

Array heights is a mountain if there exists an index i such that:

For all 0 < j <= i, heights[j - 1] <= heights[j] For all i <= k < n - 1, heights[k + 1] <= heights[k]

Return the maximum possible sum of heights of a beautiful configuration of towers. Example 1: Input: maxHeights = [5,3,4,1,1] Output: 13 Explanation: One beautiful configuration with a maximum sum is heights = [5,3,3,1,1]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 0. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 13. Example 2: Input: maxHeights = [6,5,3,9,2,7] Output: 22 Explanation: One beautiful configuration with a maximum sum is heights = [3,3,3,9,2,2]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 3. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 22. Example 3: Input: maxHeights = [3,2,5,5,2,3] Output: 18 Explanation: One beautiful configuration with a maximum sum is heights = [2,2,5,5,2,2]. This configuration is beautiful since: - 1 <= heights[i] <= maxHeights[i] - heights is a mountain of peak i = 2. Note that, for this configuration, i = 3 can also be considered a peak. It can be shown that there exists no other beautiful configuration with a sum of heights greater than 18.

Constraints:

1 <= n == maxHeights <= 105 1 <= maxHeights[i] <= 109

==

Problem Number: 1674 URL: https://leetcode.com/problems/median-of-two-sorted-arrays Title: 4. Median of Two Sorted Arrays Problem Description: Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays. The overall run time complexity should be O(log (m+n)). Example 1: Input: nums1 = [1,3], nums2 = [2] Output: 2.00000 Explanation: merged array = [1,2,3] and median is 2.

Example 2: Input: nums1 = [1,2], nums2 = [3,4] Output: 2.50000 Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Constraints:

nums1.length == m nums2.length == n 0 <= m <= 1000 0 <= n <= 1000 1 <= m + n <= 2000 -106 <= nums1[i], nums2[i] <= 106

==

Problem Number: 1675 URL: https://leetcode.com/problems/regular-expression-matching Title: 10. Regular Expression Matching Problem Description: Given an input string s and a pattern p, implement regular expression matching with support for '.' and '*' where:

'.' Matches any single character. '*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial). Example 1: Input: s = "aa", p = "a" Output: false Explanation: "a" does not match the entire string ”aa”.

Example 2: Input: s = ”aa”, p = ”a*” Output: true Explanation: ’*’ means zero or more of the preceding element, ’a’. Therefore, by repeating ’a’ once, it becomes ”aa”.

Example 3: Input: s = ”ab”, p = ”.*” Output: true Explanation: ”.*” means ”zero or more (*) of any character (.)”.

Constraints:

1 <= s.length <= 20 1 <= p.length <= 20 s contains only lowercase English letters. p contains only lowercase English letters, ’.’, and ’*’. It is guaranteed for each appearance of the character ’*’, there will be a previous valid character to match.

==

Problem Number: 1676 URL: https://leetcode.com/problems/merge-k-sorted-lists Title: 23. Merge k Sorted Lists Problem Description: You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. Merge all the linked-lists into one sorted linked-list and return it. Example 1: Input: lists = [[1,4,5],[1,3,4],[2,6]] Output: [1,1,2,3,4,4,5,6] Explanation: The linked-lists are: [1->4->5, 1->3->4, 2->6] merging them into one sorted list: 1->1->2->3->4->4->5->6

Example 2: Input: lists = [] Output: []

Example 3: Input: lists = [[]] Output: []

Constraints:

k == lists.length 0 <= k <= 104 0 <= lists[i].length <= 500 -104 <= lists[i][j] <= 104 lists[i] is sorted in ascending order. The sum of lists[i].length will not exceed 104.

==

Problem Number: 1677 URL: https://leetcode.com/problems/reverse-nodes-in-k-group Title: 25. Reverse Nodes in k-Group Problem Description: Given the head of a linked list, reverse the nodes of the list k at a time, and return the modified list. k is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of k then left-out nodes, in the end, should remain as it is. You may not alter the values in the list’s nodes, only nodes themselves may be changed. Example 1:

Input: head = [1,2,3,4,5], k = 2 Output: [2,1,4,3,5]

Example 2:

Input: head = [1,2,3,4,5], k = 3 Output: [3,2,1,4,5]

Constraints:

The number of nodes in the list is n. 1 <= k <= n <= 5000 0 <= Node.val <= 1000

Follow-up: Can you solve the problem in O(1) extra memory space?

==
Problem Number: 1678 URL: https://leetcode.com/problems/substring-with-concatenation-of-all-words Title: 30. Substring with Concatenation of All Words Problem Description: You are given a string s and an array of strings words. All the strings of words are of the same length. A concatenated substring in s is a substring that contains all the strings of any permutation of words concatenated.

For example, if words = ["ab","cd","ef"], then "abcdef", "abefcd", "cdabef", "cdefab", "efabcd", and "efcdab" are all concatenated strings. "acdbef" is not a concatenated substring because it is not the concatenation of any permutation of words.

Return the starting indices of all the concatenated substrings in s. You can return the answer in any order. Example 1: Input: s = "barfoothefoobarman", words = ["foo","bar"] Output: [0,9] Explanation: Since words.length == 2 and words[i].length == 3, the concatenated substring has to be of length 6. The substring starting at 0 is "barfoo". It is the concatenation of ["bar","foo"] which is a permutation of words. The substring starting at 9 is "foobar". It is the concatenation of ["foo","bar"] which is a permutation of words. The output order does not matter. Returning [9,0] is fine too.

Example 2: Input: s = "wordgoodgoodgoodbestword", words = ["word","good","best","word"] Output: [] Explanation: Since words.length == 4 and words[i].length == 4, the concatenated substring has to be of length 16. There is no substring of length 16 in s that is equal to the concatenation of any permutation of words. We return an empty array.

Example 3: Input: s = "barfoofoobarthefoobarman", words = ["bar","foo","the"] Output: [6,9,12] Explanation: Since words.length == 3 and words[i].length == 3, the concatenated substring has to be of length 9. The substring starting at 6 is "foobarthe". It is the concatenation of ["foo","bar","the"] which is a permutation of words. The substring starting at 9 is "barthefoo". It is the concatenation of ["bar","the","foo"] which is a permutation of words. The substring starting at 12 is "thefoobar". It is the concatenation of ["the","foo","bar"] which is a permutation of words.

Constraints:

1 <= s.length <= 104 1 <= words.length <= 5000 1 <= words[i].length <= 30 s and words[i] consist of lowercase English letters.

==
Problem Number: 1679 URL: https://leetcode.com/problems/longest-valid-parentheses Title: 32. Longest Valid Parentheses Problem Description: Given

a string containing just the characters '(' and ')', return the length of the longest valid (well-formed) parentheses substring. Example 1: Input: s = "(()" Output: 2 Explanation: The longest valid parentheses substring is "()".

Example 2: Input: s = ")()())" Output: 4 Explanation: The longest valid parentheses substring is "()()".

Example 3: Input: s = "" Output: 0

Constraints:

0 <= s.length <= 3 * 104 s[i] is '(', or ')'.

==
Problem Number: 1680 URL: https://leetcode.com/problems/sudoku-solver Title: 37. Sudoku Solver Problem Description: Write a program to solve a Sudoku puzzle by filling the empty cells. A sudoku solution must satisfy all of the following rules:

Each of the digits 1-9 must occur exactly once in each row. Each of the digits 1-9 must occur exactly once in each column. Each of the digits 1-9 must occur exactly once in each of the 9 3x3 sub-boxes of the grid.

The '.' character indicates empty cells. Example 1:

Input: board = [["5","3",".",".","7",".",".",".","."],["6",".",".","1","9","5",".",".","."],[".","9","8",".",".",".",".","6","."],["8
Output: [["5","3","4","6","7","8","9","1","2"],["6","7","2","1","9","5","3","4","8"],["1","9","8","3","4","2","5
Explanation: The input board is shown above and the only valid solution is shown below:

Constraints:

board.length == 9 board[i].length == 9 board[i][j] is a digit or '.'. It is guaranteed that the input board has only one solution.

==
Problem Number: 1681 URL: https://leetcode.com/problems/first-missing-positive Title: 41. First Missing Positive Problem Description: Given an unsorted integer array nums, return the smallest missing positive integer. You must implement an algorithm that runs in O(n) time and uses O(1) auxiliary space. Example 1: Input: nums = [1,2,0] Output: 3 Explanation: The numbers in the range [1,2] are all in the array.

Example 2: Input: nums = [3,4,-1,1] Output: 2 Explanation: 1 is in the array but 2 is missing.

Example 3: Input: nums = [7,8,9,11,12] Output: 1 Explanation: The smallest positive integer 1 is missing.

Constraints:

1 <= nums.length <= 105 -231 <= nums[i] <= 231 - 1

== Problem Number: 1682 URL: https://leetcode.com/problems/trapping-rain-water Title: 42. Trapping Rain Water Problem Description: Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining. Example 1:

Input: height = [0,1,0,2,1,0,1,3,2,1,2,1] Output: 6 Explanation: The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

Example 2: Input: height = [4,2,0,3,2,5] Output: 9

Constraints:

n == height.length 1 <= n <= 2 * 104 0 <= height[i] <= 105

== Problem Number: 1683 URL: https://leetcode.com/problems/wildcard-matching Title: 44. Wildcard Matching Problem Description: Given an input string (s) and a pattern (p), implement wildcard pattern matching with support for '?' and '*' where:

'?' Matches any single character. '*' Matches any sequence of characters (including the empty sequence).

The matching should cover the entire input string (not partial). Example 1: Input: s = "aa", p = "a" Output: false Explanation: "a" does not match the entire string "aa".

Example 2: Input: s = "aa", p = "*" Output: true Explanation: '*' matches any sequence.

Example 3: Input: s = "cb", p = "?a" Output: false Explanation: '?' matches 'c', but the second letter is 'a', which does not match 'b'.

Constraints:

0 <= s.length, p.length <= 2000 s contains only lowercase English letters. p contains only lowercase English letters, '?' or '*'.

== Problem Number: 1684 URL: https://leetcode.com/problems/n-queens Title: 51. N-Queens Problem Description: The n-queens puzzle is the problem of placing n queens on an n x n chessboard such that no two queens attack each other. Given an integer n, return all distinct solutions to the n-queens puzzle. You may return the answer in any order. Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.' both indicate a queen and an empty space, respectively. Example 1:

Input: n = 4 Output: [[".Q..","...Q","Q...","..Q."],["..Q.","Q...","...Q",".Q.."]] Explanation: There exist two distinct solutions to the 4-queens puzzle as shown above

Example 2: Input: n = 1 Output: [["Q"]]

Constraints:

1 <= n <= 9

==
Problem Number: 1685 URL: https://leetcode.com/problems/n-queens-ii Title: 52. N-Queens II Problem Description: The n-queens puzzle is the problem of placing n queens on an n x n chessboard such that no two queens attack each other. Given an integer n, return the number of distinct solutions to the n-queens puzzle. Example 1:

Input: n = 4 Output: 2 Explanation: There are two distinct solutions to the 4-queens puzzle as shown.

Example 2: Input: n = 1 Output: 1

Constraints:

1 <= n <= 9

==
Problem Number: 1686 URL: https://leetcode.com/problems/permutation-sequence Title: 60. Permutation Sequence Problem Description: The set [1, 2, 3, ..., n] contains a total of n! unique permutations. By listing and labeling all of the permutations in order, we get the following sequence for n = 3:

"123" "132" "213" "231" "312" "321"

Given n and k, return the kth permutation sequence. Example 1: Input: n = 3, k = 3 Output: "213" Example 2: Input: n = 4, k = 9 Output: "2314" Example 3: Input: n = 3, k = 1 Output: "123"

Constraints:

1 <= n <= 9 1 <= k <= n!

==
Problem Number: 1687 URL: https://leetcode.com/problems/valid-number Title: 65. Valid Number Problem Description: A valid number can be split up into these components (in order):

A decimal number or an integer. (Optional) An 'e' or 'E', followed by an integer.

A decimal number can be split up into these components (in order):

(Optional) A sign character (either '+' or '-'). One of the following formats:

One or more digits, followed by a dot '.'. One or more digits, followed by a dot '.', followed by one or more digits. A dot '.', followed by one or more digits.

An integer can be split up into these components (in order):

(Optional) A sign character (either '+' or '-'). One or more digits.

For example, all the following are valid numbers: ["2", "0089", "-0.1", "+3.14", "4.", "-.9", "2e10", "-90E3", "3e+7", "+6e-1", "53.5e93", "-123.456e789"], while the following are not valid numbers: ["abc", "1a", "1e", "e3", "99e2.5", "--6", "-+3", "95a54e53"]. Given a string s, return true if s is a valid number. Example 1: Input: s = "0" Output: true

Example 2: Input: s = "e" Output: false

Example 3: Input: s = "." Output: false

Constraints:

1 <= s.length <= 20 s consists of only English letters (both uppercase and lowercase), digits (0-9), plus '+', minus '-', or dot '.'.

==

Problem Number: 1688 URL: https://leetcode.com/problems/text-justification Title: 68. Text Justification Problem Description: Given an array of strings words and a width maxWidth, format the text such that each line has exactly maxWidth characters and is fully (left and right) justified. You should pack your words in a greedy approach; that is, pack as many words as you can in each line. Pad extra spaces ' ' when necessary so that each line has exactly maxWidth characters. Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line does not divide evenly between words, the empty slots on the left will be assigned more spaces than the slots on the right. For the last line of text, it should be left-justified, and no extra space is inserted between words. Note:

A word is defined as a character sequence consisting of non-space characters only. Each word's length is guaranteed to be greater than 0 and not exceed maxWidth. The input array words contains at least one word.

Example 1: Input: words = ["This", "is", "an", "example", "of", "text", "justification."], maxWidth = 16 Output: ["This is an", "example of text", "justification. "] Example 2: Input: words = ["What","must","be","acknowledgment","shall","be"], maxWidth = 16 Output: ["What must be", "acknowledgment ", "shall be "] Explanation: Note that the last line is "shall be " instead of "shall be", because the last line must be left-justified instead of fully-justified. Note that the second line is also left-justified because it contains only one word. Example 3: Input: words = ["Science","is","what","we","understand","well","enough","to","explain","to","a","computer.","Art","is","everythi
maxWidth = 20 Output: ["Science is what we", "understand well", "enough to explain to", "a computer. Art is", "everything else we", "do "] Constraints:

1 <= words.length <= 300 1 <= words[i].length <= 20 words[i] consists of only English letters and symbols. 1 <= maxWidth <= 100 words[i].length <= maxWidth

==

Problem Number: 1689 URL: https://leetcode.com/problems/minimum-window-substring Title: 76. Minimum Window Substring Problem Description: Given two strings s and t of lengths m and n respectively, return the minimum window substring of s such that every character in t (including duplicates) is included in the window. If there is no such substring, return the empty string "". The testcases will be generated such that the answer is unique. Example 1: Input: s = "ADOBECODEBANC", t = "ABC" Output: "BANC" Explanation: The minimum window substring "BANC" includes 'A', 'B', and 'C' from string t.

Example 2: Input: s = "a", t = "a" Output: "a" Explanation: The entire string s is the minimum window.

Example 3: Input: s = "a", t = "aa" Output: "" Explanation: Both 'a's from t must be included in the window. Since the largest window of s only has one 'a', return empty string.

Constraints:

m == s.length n == t.length 1 <= m, n <= 105 s and t consist of uppercase and lowercase English letters.

Follow up: Could you find an algorithm that runs in O(m + n) time?

==

Problem Number: 1690 URL: https://leetcode.com/problems/largest-rectangle-in-histogram Title: 84. Largest Rectangle in Histogram Problem Description: Given an array of integers heights representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram. Example 1:

Input: heights = [2,1,5,6,2,3] Output: 10 Explanation: The above is a histogram where width of each bar is 1. The largest rectangle is shown in the red area, which has an area = 10 units.

Example 2:

Input: heights = [2,4] Output: 4

Constraints:

1 <= heights.length <= 105 0 <= heights[i] <= 104

==

Problem Number: 1691 URL: https://leetcode.com/problems/maximal-rectangle Title: 85. Maximal Rectangle Problem Description: Given a rows x cols binary matrix filled with 0's and 1's, find the largest rectangle containing only 1's and return its area. Example 1:

Input: matrix = [["1","0","1","0","0"],["1","0","1","1","1"],["1","1","1","1","1"],["1","0","0","1","0"]] Output: 6 Explanation: The maximal rectangle is shown in the above picture.

Example 2: Input: matrix = [["0"]] Output: 0

Example 3: Input: matrix = [["1"]] Output: 1

Constraints:

rows == matrix.length cols == matrix[i].length 1 <= row, cols <= 200 matrix[i][j] is '0' or '1'.

==
Problem Number: 1692 URL: https://leetcode.com/problems/scramble-string Title: 87. Scramble String Problem Description: We can scramble a string s to get a string t using the following algorithm:

If the length of the string is 1, stop. If the length of the string is > 1, do the following:

Split the string into two non-empty substrings at a random index, i.e., if the string is s, divide it to x and y where s = x + y. Randomly decide to swap the two substrings or to keep them in the same order. i.e., after this step, s may become s = x + y or s = y + x. Apply step 1 recursively on each of the two substrings x and y.

Given two strings s1 and s2 of the same length, return true if s2 is a scrambled string of s1, otherwise, return false. Example 1: Input: s1 = "great", s2 = "rgeat" Output: true Explanation: One possible scenario applied on s1 is: "great" --> "gr/eat" // divide at random index. "gr/eat" --> "gr/eat" // random decision is not to swap the two substrings and keep them in order. "gr/eat" --> "g/r / e/at" // apply the same algorithm recursively on both substrings. divide at random index each of them. "g/r / e/at" --> "r/g / e/at" // random decision was to swap the first substring and to keep the second substring in the same order. "r/g / e/at" --> "r/g / e/ a/t" // again apply the algorithm recursively, divide "at" to "a/t". "r/g / e/ a/t" --> "r/g / e/ a/t" // random decision is to keep both substrings in the same order. The algorithm stops now, and the result string is "rgeat" which is s2. As one possible scenario led s1 to be scrambled to s2, we return true.

Example 2: Input: s1 = "abcde", s2 = "caebd" Output: false

Example 3: Input: s1 = "a", s2 = "a" Output: true

Constraints:

s1.length == s2.length 1 <= s1.length <= 30 s1 and s2 consist of lowercase English letters.

==
Problem Number: 1693 URL: https://leetcode.com/problems/distinct-subsequences Title: 115. Distinct Subsequences Problem Description: Given two strings s and t, return the number of distinct subsequences of s which equals t. The test cases are generated so that the answer fits on a 32-bit signed integer. Example 1: Input: s = "rabbbit", t = "rabbit" Output: 3

Explanation: As shown below, there are 3 ways you can generate "rabbit" from s. rabbbit rabbbit rabbbit

Example 2: Input: s = "babgbag", t = "bag" Output: 5 Explanation: As shown below, there are 5 ways you can generate "bag" from s. babgbag babgbag babgbag babgbag babgbag Constraints:

1 <= s.length, t.length <= 1000 s and t consist of English letters.

==

Problem Number: 1694 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iii Title: 123. Best Time to Buy and Sell Stock III Problem Description: You are given an array prices where prices[i] is the price of a given stock on the ith day. Find the maximum profit you can achieve. You may complete at most two transactions. Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again). Example 1: Input: prices = [3,3,5,0,0,3,1,4] Output: 6 Explanation: Buy on day 4 (price = 0) and sell on day 6 (price = 3), profit = 3-0 = 3. Then buy on day 7 (price = 1) and sell on day 8 (price = 4), profit = 4-1 = 3. Example 2: Input: prices = [1,2,3,4,5] Output: 4 Explanation: Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4. Note that you cannot buy on day 1, buy on day 2 and sell them later, as you are engaging multiple transactions at the same time. You must sell before buying again.

Example 3: Input: prices = [7,6,4,3,1] Output: 0 Explanation: In this case, no transaction is done, i.e. max profit = 0.

Constraints:

1 <= prices.length <= 105 0 <= prices[i] <= 105

==

Problem Number: 1695 URL: https://leetcode.com/problems/binary-tree-maximum-path-sum Title: 124. Binary Tree Maximum Path Sum Problem Description: A path in a binary tree is a sequence of nodes where each pair of adjacent nodes in the sequence has an edge connecting them. A node can only appear in the sequence at most once. Note that the path does not need to pass through the root. The path sum of a path is the sum of the node's values in the path. Given the root of a binary tree, return the maximum path sum of any non-empty path. Example 1:

Input: root = [1,2,3] Output: 6 Explanation: The optimal path is 2 -> 1 -> 3 with a path sum of 2 + 1 + 3 = 6.

Example 2:

Input: root = [-10,9,20,null,null,15,7] Output: 42 Explanation: The optimal path is 15 -> 20 -> 7 with a path sum of 15 + 20 + 7 = 42.

Constraints:

The number of nodes in the tree is in the range [1, 3 * 104]. -1000 <= Node.val <= 1000

===
Problem Number: 1696 URL: https://leetcode.com/problems/word-ladder-ii Title: 126. Word Ladder II Problem Description: A transformation sequence from word beginWord to word endWord using a dictionary wordList is a sequence of words beginWord -> s1 -> s2 -> ... -> sk such that:

Every adjacent pair of words differs by a single letter. Every si for 1 <= i <= k is in wordList. Note that beginWord does not need to be in wordList. sk == endWord

Given two words, beginWord and endWord, and a dictionary wordList, return all the shortest transformation sequences from beginWord to endWord, or an empty list if no such sequence exists. Each sequence should be returned as a list of the words [beginWord, s1, s2, ..., sk]. Example 1: Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log","cog"] Output: [["hit","hot","dot","dog","cog"],["hit","hot","lot","log","cog"]] Explanation: There are 2 shortest transformation sequences: "hit" -> "hot" -> "dot" -> "dog" -> "cog" "hit" -> "hot" -> "lot" -> "log" -> "cog"

Example 2: Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log"] Output: [] Explanation: The endWord "cog" is not in wordList, therefore there is no valid transformation sequence.

Constraints:

1 <= beginWord.length <= 5 endWord.length == beginWord.length 1 <= wordList.length <= 500 wordList[i].length == beginWord.length beginWord, endWord, and wordList[i] consist of lowercase English letters. beginWord != endWord All the words in wordList are unique. The sum of all shortest transformation sequences does not exceed 105.

===
Problem Number: 1697 URL: https://leetcode.com/problems/word-ladder Title: 127. Word Ladder Problem Description: A transformation sequence from word beginWord to word endWord using a dictionary wordList is a sequence of words beginWord -> s1 -> s2 -> ... -> sk such that:

Every adjacent pair of words differs by a single letter. Every si for 1 <= i <= k is in wordList. Note that beginWord does not need to be in wordList. sk == endWord

Given two words, beginWord and endWord, and a dictionary wordList, return the number of words in the shortest transformation sequence from beginWord to endWord, or 0 if no such sequence exists. Example 1: Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log","cog"] Output: 5 Explanation: One shortest transformation sequence is "hit" -> "hot" -> "dot" -> "dog" -> cog", which is 5 words long.

Example 2: Input: beginWord = "hit", endWord = "cog", wordList = ["hot","dot","dog","lot","log"] Output: 0 Explanation: The endWord "cog" is not in wordList, therefore there is no valid transformation sequence.

Constraints:

1 <= beginWord.length <= 10 endWord.length == beginWord.length 1 <= wordList.length <= 5000 wordList[i].length == beginWord.length beginWord, endWord, and wordList[i] consist of lowercase English letters. beginWord != endWord All the words in wordList are unique.

==
Problem Number: 1698 URL: https://leetcode.com/problems/palindrome-partitioning-ii Title: 132. Palindrome Partitioning II Problem Description: Given a string s, partition s such that every substring of the partition is a palindrome. Return the minimum cuts needed for a palindrome partitioning of s. Example 1: Input: s = "aab" Output: 1 Explanation: The palindrome partitioning ["aa","b"] could be produced using 1 cut.

Example 2: Input: s = "a" Output: 0

Example 3: Input: s = "ab" Output: 1

Constraints:

1 <= s.length <= 2000 s consists of lowercase English letters only.

==
Problem Number: 1699 URL: https://leetcode.com/problems/candy Title: 135. Candy Problem Description: There are n children standing in a line. Each child is assigned a rating value given in the integer array ratings. You are giving candies to these children subjected to the following requirements:

Each child must have at least one candy. Children with a higher rating get more candies than their neighbors.

Return the minimum number of candies you need to have to distribute the candies to the children. Example 1: Input: ratings = [1,0,2] Output: 5 Explanation: You can allocate to the first, second and third child with 2, 1, 2 candies respectively.

Example 2: Input: ratings = [1,2,2] Output: 4 Explanation: You can allocate to the first, second and third child with 1, 2, 1 candies respectively. The third child gets 1 candy because it satisfies the above two conditions.

Constraints:

n == ratings.length 1 <= n <= 2 * 104 0 <= ratings[i] <= 2 * 104

==
Problem Number: 1700 URL: https://leetcode.com/problems/word-break-ii Title: 140. Word Break II Problem Description: Given a string s and a dictionary of strings wordDict, add spaces in s to construct a sentence where

each word is a valid dictionary word. Return all such possible sentences in any order. Note that the same word in the dictionary may be reused multiple times in the segmentation. Example 1: Input: s = "catsanddog", wordDict = ["cat","cats","and","sand","dog"] Output: ["cats and dog","cat sand dog"]

Example 2: Input: s = "pineapplepenapple", wordDict = ["apple","pen","applepen","pine","pineapple"] Output: ["pine apple pen apple","pineapple pen apple","pine applepen apple"] Explanation: Note that you are allowed to reuse a dictionary word.

Example 3: Input: s = "catsandog", wordDict = ["cats","dog","sand","and","cat"] Output: []

Constraints:

1 <= s.length <= 20 1 <= wordDict.length <= 1000 1 <= wordDict[i].length <= 10 s and wordDict[i] consist of only lowercase English letters. All the strings of wordDict are unique. Input is generated in a way that the length of the answer doesn't exceed 105.

==
Problem Number: 1701 URL: https://leetcode.com/problems/max-points-on-a-line Title: 149. Max Points on a Line Problem Description: Given an array of points where points[i] = [xi, yi] represents a point on the X-Y plane, return the maximum number of points that lie on the same straight line. Example 1:

Input: points = [[1,1],[2,2],[3,3]] Output: 3

Example 2:

Input: points = [[1,1],[3,2],[5,3],[4,1],[2,3],[1,4]] Output: 4

Constraints:

1 <= points.length <= 300 points[i].length == 2 -104 <= xi, yi <= 104 All the points are unique.

==
Problem Number: 1702 URL: https://leetcode.com/problems/find-minimum-in-rotated-sorted-array-ii Title: 154. Find Minimum in Rotated Sorted Array II Problem Description: Suppose an array of length n sorted in ascending order is rotated between 1 and n times. For example, the array nums = [0,1,4,4,5,6,7] might become:

[4,5,6,7,0,1,4] if it was rotated 4 times. [0,1,4,4,5,6,7] if it was rotated 7 times.

Notice that rotating an array [a[0], a[1], a[2], ..., a[n-1]] 1 time results in the array [a[n-1], a[0], a[1], a[2], ..., a[n-2]]. Given the sorted rotated array nums that may contain duplicates, return the minimum element of this array. You must decrease the overall operation steps as much as possible. Example 1: Input: nums = [1,3,5] Output: 1 Example 2: Input: nums = [2,2,2,0,1] Output: 0

Constraints:

n == nums.length 1 <= n <= 5000 -5000 <= nums[i] <= 5000 nums is sorted and rotated between 1 and n times.

Follow up: This problem is similar to Find Minimum in Rotated Sorted Array, but nums may contain duplicates. Would this affect the runtime complexity? How and why?

==
Problem Number: 1703 URL: https://leetcode.com/problems/maximum-gap Title: 164. Maximum Gap Problem Description: Given an integer array nums, return the maximum difference between two successive elements in its sorted form. If the array contains less than two elements, return 0. You must write an algorithm that runs in linear time and uses linear extra space. Example 1: Input: nums = [3,6,9,1] Output: 3 Explanation: The sorted form of the array is [1,3,6,9], either (3,6) or (6,9) has the maximum difference 3.

Example 2: Input: nums = [10] Output: 0 Explanation: The array contains less than 2 elements, therefore return 0.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==
Problem Number: 1704 URL: https://leetcode.com/problems/dungeon-game Title: 174. Dungeon Game Problem Description: The demons had captured the princess and imprisoned her in the bottom-right corner of a dungeon. The dungeon consists of m x n rooms laid out in a 2D grid. Our valiant knight was initially positioned in the top-left room and must fight his way through dungeon to rescue the princess. The knight has an initial health point represented by a positive integer. If at any point his health point drops to 0 or below, he dies immediately. Some of the rooms are guarded by demons (represented by negative integers), so the knight loses health upon entering these rooms; other rooms are either empty (represented as 0) or contain magic orbs that increase the knight's health (represented by positive integers). To reach the princess as quickly as possible, the knight decides to move only rightward or downward in each step. Return the knight's minimum initial health so that he can rescue the princess. Note that any room can contain threats or power-ups, even the first room the knight enters and the bottom-right room where the princess is imprisoned. Example 1:

Input: dungeon = [[-2,-3,3],[-5,-10,1],[10,30,-5]] Output: 7 Explanation: The initial health of the knight must be at least 7 if he follows the optimal path: RIGHT-> RIGHT -> DOWN -> DOWN.

Example 2: Input: dungeon = [[0]] Output: 1

Constraints:

m == dungeon.length n == dungeon[i].length 1 <= m, n <= 200 -1000 <= dungeon[i][j] <= 1000

==
Problem Number: 1705 URL: https://leetcode.com/problems/best-time-to-buy-and-sell-stock-iv Title: 188. Best Time to Buy and Sell Stock IV Problem Description: You are given an integer array prices where prices[i] is the price of a given stock on the ith day, and an integer k. Find the maximum profit you can achieve. You may complete at most k transactions: i.e. you may buy at most k times and sell at most k times. Note: You may not engage in multiple transactions simultaneously (i.e., you must sell the stock before you buy again). Example 1: Input: k = 2, prices = [2,4,1] Output: 2 Explanation: Buy on day 1 (price = 2) and sell on day 2 (price = 4), profit = 4-2 = 2.

Example 2: Input: k = 2, prices = [3,2,6,5,0,3] Output: 7 Explanation: Buy on day 2 (price = 2) and sell on day 3 (price = 6), profit = 6-2 = 4. Then buy on day 5 (price = 0) and sell on day 6 (price = 3), profit = 3-0 = 3.

Constraints:

1 <= k <= 100 1 <= prices.length <= 1000 0 <= prices[i] <= 1000

==
Problem Number: 1706 URL: https://leetcode.com/problems/word-search-ii Title: 212. Word Search II Problem Description: Given an m x n board of characters and a list of strings words, return all words on the board. Each word must be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once in a word. Example 1:

Input: board = [["o","a","a","n"],["e","t","a","e"],["i","h","k","r"],["i","f","l","v"]], words = ["oath","pea","eat","rain"] Output: ["eat","oath"]

Example 2:

Input: board = [["a","b"],["c","d"]], words = ["abcb"] Output: []

Constraints:

m == board.length n == board[i].length 1 <= m, n <= 12 board[i][j] is a lowercase English letter. 1 <= words.length <= 3 * 104 1 <= words[i].length <= 10 words[i] consists of lowercase English letters. All the strings of words are unique.

==
Problem Number: 1707 URL: https://leetcode.com/problems/shortest-palindrome Title: 214. Shortest Palindrome Problem Description: You are given a string s. You can convert s to a palindrome by adding characters in front of it. Return the shortest palindrome you can find by performing this transformation. Example 1: Input: s = "aacecaaa" Output: "aaacecaaa" Example 2: Input: s = "abcd" Output: "dcbabcd"

Constraints:

0 <= s.length <= 5 * 104 s consists of lowercase English letters only.

==

Problem Number: 1708 URL: https://leetcode.com/problems/the-skyline-problem Title: 218. The Skyline Problem Problem Description: A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Given the locations and heights of all the buildings, return the skyline formed by these buildings collectively. The geometric information of each building is given in the array buildings where buildings[i] = [lefti, righti, heighti]:

lefti is the x coordinate of the left edge of the ith building. righti is the x coordinate of the right edge of the ith building. heighti is the height of the ith building.

You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0. The skyline should be represented as a list of "key points" sorted by their x-coordinate in the form [[x1,y1],[x2,y2],...]. Each key point is the left endpoint of some horizontal segment in the skyline except the last point in the list, which always has a y-coordinate 0 and is used to mark the skyline's termination where the rightmost building ends. Any ground between the leftmost and rightmost buildings should be part of the skyline's contour. Note: There must be no consecutive horizontal lines of equal height in the output skyline. For instance, [...,[2 3],[4 5],[7 5],[11 5],[12 7],...] is not acceptable; the three lines of height 5 should be merged into one in the final output as such: [...,[2 3],[4 5],[12 7],...] Example 1:

Input: buildings = [[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]] Output: [[2,10],[3,15],[7,12],[12,0],[15,10],[20,8],[24,0]] Explanation: Figure A shows the buildings of the input. Figure B shows the skyline formed by those buildings. The red points in figure B represent the key points in the output list.

Example 2: Input: buildings = [[0,2,3],[2,5,3]] Output: [[0,3],[5,0]]

Constraints:

1 <= buildings.length <= 104 0 <= lefti < righti <= 231 - 1 1 <= heighti <= 231 - 1 buildings is sorted by lefti in non-decreasing order.

==

Problem Number: 1709 URL: https://leetcode.com/problems/contains-duplicate-iii Title: 220. Contains Duplicate III Problem Description: You are given an integer array nums and two integers indexDiff and valueDiff. Find a pair of indices (i, j) such that:

i != j, abs(i - j) <= indexDiff. abs(nums[i] - nums[j]) <= valueDiff, and

Return true if such pair exists or false otherwise. Example 1: Input: nums = [1,2,3,1], indexDiff = 3, valueDiff = 0 Output: true Explanation: We can choose (i, j) = (0, 3). We satisfy the three conditions: i != j --> 0 != 3 abs(i - j) <= indexDiff --> abs(0 - 3) <= 3 abs(nums[i] - nums[j]) <= valueDiff --> abs(1 - 1) <= 0

Example 2: Input: nums = [1,5,9,1,5,9], indexDiff = 2, valueDiff = 3 Output: false Explanation: After trying all the possible pairs (i, j), we cannot satisfy the three conditions, so we return false.

Constraints:

2 <= nums.length <= 105 -109 <= nums[i] <= 109 1 <= indexDiff <= nums.length 0 <= valueDiff <= 109

==
Problem Number: 1710 URL: https://leetcode.com/problems/basic-calculator Title: 224. Basic Calculator Problem Description: Given a string s representing a valid expression, implement a basic calculator to evaluate it, and return the result of the evaluation. Note: You are not allowed to use any built-in function which evaluates strings as mathematical expressions, such as eval(). Example 1: Input: s = "1 + 1" Output: 2

Example 2: Input: s = " 2-1 + 2 " Output: 3

Example 3: Input: s = "(1+(4+5+2)-3)+(6+8)" Output: 23

Constraints:

1 <= s.length <= 3 * 105 s consists of digits, '+', '-', '(', ')', and ' '. s represents a valid expression. '+' is not used as a unary operation (i.e., "+1" and "+(2 + 3)" is invalid). '-' could be used as a unary operation (i.e., "-1" and "-(2 + 3)" is valid). There will be no two consecutive operators in the input. Every number and running calculation will fit in a signed 32-bit integer.

==
Problem Number: 1711 URL: https://leetcode.com/problems/number-of-digit-one Title: 233. Number of Digit One Problem Description: Given an integer n, count the total number of digit 1 appearing in all non-negative integers less than or equal to n. Example 1: Input: n = 13 Output: 6

Example 2: Input: n = 0 Output: 0

Constraints:

0 <= n <= 109

==
Problem Number: 1712 URL: https://leetcode.com/problems/sliding-window-maximum Title: 239. Sliding Window Maximum Problem Description: You are given an array of integers nums, there is a sliding window of size k which is moving from the very left of the array to the very right. You can only see the k numbers in the window. Each time the sliding window moves right by one position. Return the max sliding window. Example 1: Input: nums = [1,3,-1,-3,5,3,6,7], k = 3 Output: [3,3,5,5,6,7] Explanation: Window position Max --------------- ----- [1 3 -1] -3 5 3 6 7 3 1 [3 -1 -3] 5 3 6 7 3 1 3 [-1 -3 5] 3 6 7 5 1 3 -1 [-3 5 3] 6 7 5 1 3 -1 -3 [5 3 6] 7 6 1 3 -1 -3 5 [3 6 7] 7

Example 2: Input: nums = [1], k = 1 Output: [1]

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104 1 <= k <= nums.length

==
Problem Number: 1713 URL: https://leetcode.com/problems/integer-to-english-words Title: 273. Integer to English Words Problem Description: Convert a non-negative integer num to its English words representation. Example 1: Input: num = 123 Output: "One Hundred Twenty Three"

Example 2: Input: num = 12345 Output: "Twelve Thousand Three Hundred Forty Five"

Example 3: Input: num = 1234567 Output: "One Million Two Hundred Thirty Four Thousand Five Hundred Sixty Seven"

Constraints:

0 <= num <= 231 - 1

==
Problem Number: 1714 URL: https://leetcode.com/problems/expression-add-operators Title: 282. Expression Add Operators Problem Description: Given a string num that contains only digits and an integer target, return all possibilities to insert the binary operators '+', '-', and/or '*' between the digits of num so that the resultant expression evaluates to the target value. Note that operands in the returned expressions should not contain leading zeros. Example 1: Input: num = "123", target = 6 Output: ["1*2*3","1+2+3"] Explanation: Both "1*2*3" and "1+2+3" evaluate to 6.

Example 2: Input: num = "232", target = 8 Output: ["2*3+2","2+3*2"] Explanation: Both "2*3+2" and "2+3*2" evaluate to 8.

Example 3: Input: num = "3456237490", target = 9191 Output: [] Explanation: There are no expressions that can be created from "3456237490" to evaluate to 9191.

Constraints:

1 <= num.length <= 10 num consists of only digits. -231 <= target <= 231 - 1

==
Problem Number: 1715 URL: https://leetcode.com/problems/find-median-from-data-stream Title: 295. Find Median from Data Stream Problem Description: The median is the middle value in an ordered integer list. If the size of the list is even, there is no middle value, and the median is the mean of the two middle values.

For example, for arr = [2,3,4], the median is 3. For example, for arr = [2,3], the median is (2 + 3) / 2 = 2.5.

Implement the MedianFinder class:

MedianFinder() initializes the MedianFinder object. void addNum(int num) adds the integer num from the data stream to the data structure. double findMedian() returns the median of all elements so far. Answers within 10-5 of the actual answer will be accepted.

Example 1: Input ["MedianFinder", "addNum", "addNum", "findMedian", "addNum", "findMedian"] [[], [1], [2], [], [3], []] Output [null, null, null, 1.5, null, 2.0]

Explanation MedianFinder medianFinder = new MedianFinder(); medianFinder.addNum(1); // arr = [1] medianFinder.addNum(2); // arr = [1, 2] medianFinder.findMedian(); // return 1.5 (i.e., (1 + 2) / 2) medianFinder.addNum(3); // arr[1, 2, 3] medianFinder.findMedian(); // return 2.0

Constraints:

-105 <= num <= 105 There will be at least one element in the data structure before calling findMedian. At most 5 * 104 calls will be made to addNum and findMedian.

Follow up:

If all integer numbers from the stream are in the range [0, 100], how would you optimize your solution? If 99% of all integer numbers from the stream are in the range [0, 100], how would you optimize your solution?

==

Problem Number: 1716 URL: https://leetcode.com/problems/serialize-and-deserialize-binary-tree Title: 297. Serialize and Deserialize Binary Tree Problem Description: Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment. Design an algorithm to serialize and deserialize a binary tree. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that a binary tree can be serialized to a string and this string can be deserialized to the original tree structure. Clarification: The input/output format is the same as how LeetCode serializes a binary tree. You do not necessarily need to follow this format, so please be creative and come up with different approaches yourself. Example 1:

Input: root = [1,2,3,null,null,4,5] Output: [1,2,3,null,null,4,5]

Example 2: Input: root = [] Output: []

Constraints:

The number of nodes in the tree is in the range [0, 104]. -1000 <= Node.val <= 1000

==
Problem Number: 1717 URL: https://leetcode.com/problems/remove-invalid-parentheses Title: 301. Remove Invalid Parentheses Problem Description: Given a string s that contains parentheses and letters, remove the minimum number of invalid parentheses to make the input string valid. Return a list of unique strings that are valid with the minimum number of removals. You may return the answer in any order. Example 1: Input: s = ”()())()” Output: [”(())()”,”()()()”]

Example 2: Input: s = ”(a)())()” Output: [”(a())()”,”(a)()()”]

Example 3: Input: s = ”)(” Output: [””]

Constraints:

1 <= s.length <= 25 s consists of lowercase English letters and parentheses ’(’ and ’)’. There will be at most 20 parentheses in s.

==
Problem Number: 1718 URL: https://leetcode.com/problems/burst-balloons Title: 312. Burst Balloons Problem Description: You are given n balloons, indexed from 0 to n - 1. Each balloon is painted with a number on it represented by an array nums. You are asked to burst all the balloons. If you burst the ith balloon, you will get nums[i - 1] * nums[i] * nums[i + 1] coins. If i - 1 or i + 1 goes out of bounds of the array, then treat it as if there is a balloon with a 1 painted on it. Return the maximum coins you can collect by bursting the balloons wisely. Example 1: Input: nums = [3,1,5,8] Output: 167 Explanation: nums = [3,1,5,8] --> [3,5,8] --> [3,8] --> [8] --> [] coins = 3*1*5 + 3*5*8 + 1*3*8 + 1*8*1 = 167 Example 2: Input: nums = [1,5] Output: 10

Constraints:

n == nums.length 1 <= n <= 300 0 <= nums[i] <= 100

==
Problem Number: 1719 URL: https://leetcode.com/problems/count-of-smaller-numbers-after-self Title: 315. Count of Smaller Numbers After Self Problem Description: Given an integer array nums, return an integer array counts where counts[i] is the number of smaller elements to the right of nums[i]. Example 1: Input: nums = [5,2,6,1] Output: [2,1,1,0] Explanation: To the right of 5 there are 2 smaller elements (2 and 1). To the right of 2 there is only 1 smaller element (1). To the right of 6 there is 1 smaller element (1). To the right of 1 there is 0 smaller element.

Example 2: Input: nums = [-1] Output: [0]

Example 3: Input: nums = [-1,-1] Output: [0,0]

Constraints:

1 <= nums.length <= 105 -104 <= nums[i] <= 104

==

Problem Number: 1720 URL: https://leetcode.com/problems/create-maximum-number Title: 321. Create Maximum Number Problem Description: You are given two integer arrays nums1 and nums2 of lengths m and n respectively. nums1 and nums2 represent the digits of two numbers. You are also given an integer k. Create the maximum number of length k <= m + n from digits of the two numbers. The relative order of the digits from the same array must be preserved. Return an array of the k digits representing the answer. Example 1: Input: nums1 = [3,4,6,5], nums2 = [9,1,2,5,8,3], k = 5 Output: [9,8,6,5,3]

Example 2: Input: nums1 = [6,7], nums2 = [6,0,4], k = 5 Output: [6,7,6,0,4]

Example 3: Input: nums1 = [3,9], nums2 = [8,9], k = 3 Output: [9,8,9]

Constraints:

m == nums1.length n == nums2.length 1 <= m, n <= 500 0 <= nums1[i], nums2[i] <= 9 1 <= k <= m + n

==

Problem Number: 1721 URL: https://leetcode.com/problems/count-of-range-sum Title: 327. Count of Range Sum Problem Description: Given an integer array nums and two integers lower and upper, return the number of range sums that lie in [lower, upper] inclusive. Range sum S(i, j) is defined as the sum of the elements in nums between indices i and j inclusive, where i <= j. Example 1: Input: nums = [-2,5,-1], lower = -2, upper = 2 Output: 3 Explanation: The three ranges are: [0,0], [2,2], and [0,2] and their respective sums are: -2, -1, 2.

Example 2: Input: nums = [0], lower = 0, upper = 0 Output: 1

Constraints:

1 <= nums.length <= 105 -231 <= nums[i] <= 231 - 1 -105 <= lower <= upper <= 105 The answer is guaranteed to fit in a 32-bit integer.

==

Problem Number: 1722 URL: https://leetcode.com/problems/longest-increasing-path-in-a-matrix Title: 329. Longest Increasing Path in a Matrix Problem Description: Given an m x n integers matrix, return the length of the longest increasing path in matrix. From each cell, you can either move in four directions: left, right, up, or down. You may not move diagonally or move outside the boundary (i.e., wrap-around is not allowed). Example 1:

Input: matrix = [[9,9,4],[6,6,8],[2,1,1]] Output: 4 Explanation: The longest increasing path is [1, 2, 6, 9].

Example 2:

Input: matrix = [[3,4,5],[3,2,6],[2,2,1]] Output: 4 Explanation: The longest increasing path is [3, 4, 5, 6]. Moving diagonally is not allowed.

Example 3: Input: matrix = [[1]] Output: 1

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 200 0 <= matrix[i][j] <= 231 - 1

==
Problem Number: 1723 URL: https://leetcode.com/problems/patching-array Title: 330. Patching Array Problem Description: Given a sorted integer array nums and an integer n, add/patch elements to the array such that any number in the range [1, n] inclusive can be formed by the sum of some elements in the array. Return the minimum number of patches required. Example 1: Input: nums = [1,3], n = 6 Output: 1 Explanation: Combinations of nums are [1], [3], [1,3], which form possible sums of: 1, 3, 4. Now if we add/patch 2 to nums, the combinations are: [1], [2], [3], [1,3], [2,3], [1,2,3]. Possible sums are 1, 2, 3, 4, 5, 6, which now covers the range [1, 6]. So we only need 1 patch.

Example 2: Input: nums = [1,5,10], n = 20 Output: 2 Explanation: The two patches can be [2, 4].

Example 3: Input: nums = [1,2,2], n = 5 Output: 0

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 104 nums is sorted in ascending order. 1 <= n <= 231 - 1

==
Problem Number: 1724 URL: https://leetcode.com/problems/reconstruct-itinerary Title: 332. Reconstruct Itinerary Problem Description: You are given a list of airline tickets where tickets[i] = [fromi, toi] represent the departure and the arrival airports of one flight. Reconstruct the itinerary in order and return it. All of the tickets belong to a man who departs from ”JFK”, thus, the itinerary must begin with ”JFK”. If there are multiple valid itineraries, you should return the itinerary that has the smallest lexical order when read as a single string.

For example, the itinerary [”JFK”, ”LGA”] has a smaller lexical order than [”JFK”, ”LGB”].

You may assume all tickets form at least one valid itinerary. You must use all the tickets once and only once. Example 1:

Input: tickets = [[”MUC”,”LHR”],[”JFK”,”MUC”],[”SFO”,”SJC”],[”LHR”,”SFO”]] Output: [”JFK”,”MUC”,”LHR”,”SFO”,”SJC”]

Example 2:

Input: tickets = [[”JFK”,”SFO”],[”JFK”,”ATL”],[”SFO”,”ATL”],[”ATL”,”JFK”],[”ATL”,”SFO”]] Output: [”JFK”,”ATL”,”JFK”,”SFO”,”ATL”,”SFO”] Explanation: Another possible reconstruction is ["JFK","SFO","ATL","JFK","ATL","SFO"] but it is larger in lexical order.

Constraints:

1 <= tickets.length <= 300 tickets[i].length == 2 fromi.length == 3 toi.length == 3 fromi and toi consist of uppercase English letters. fromi != toi

==
Problem Number: 1725 URL: https://leetcode.com/problems/self-crossing Title: 335. Self Crossing Problem Description: You are given an array of integers distance. You start at the point (0, 0) on an X-Y plane, and you move distance[0] meters to the north, then distance[1] meters to the west, distance[2] meters to the south, distance[3] meters to the east, and so on. In other words, after each move, your direction changes counter-clockwise. Return true if your path crosses itself or false if it does not.   Example 1:

Input: distance = [2,1,1,2] Output: true Explanation: The path crosses itself at the point (0, 1).

Example 2:

Input: distance = [1,2,3,4] Output: false Explanation: The path does not cross itself at any point.

Example 3:

Input: distance = [1,1,1,2,1] Output: true Explanation: The path crosses itself at the point (0, 0).

Constraints:

1 <= distance.length <= 105 1 <= distance[i] <= 105

==
Problem Number: 1726 URL: https://leetcode.com/problems/palindrome-pairs Title: 336. Palindrome Pairs Problem Description: You are given a 0-indexed array of unique strings words. A palindrome pair is a pair of integers (i, j) such that:

0 <= i, j < words.length, i != j, and words[i] + words[j] (the concatenation of the two strings) is a palindrome.

Return an array of all the palindrome pairs of words. You must write an algorithm with O(sum of words[i].length) runtime complexity.   Example 1: Input: words = ["abcd","dcba","lls","s","sssll"] Output: [[0,1],[1,0],[3,2],[2,4]] Explanation: The palindromes are ["abcddcba","dcbaabcd","slls","llssssll"]

Example 2: Input: words = ["bat","tab","cat"] Output: [[0,1],[1,0]] Explanation: The palindromes are ["battab","tabbat"]

Example 3: Input: words = ["a",""] Output: [[0,1],[1,0]] Explanation: The palindromes are ["a","a"]

Constraints:

1 <= words.length <= 5000 0 <= words[i].length <= 300 words[i] consists of lowercase English letters.

==
Problem Number: 1727 URL: https://leetcode.com/problems/data-stream-as-disjoint-intervals Title: 352. Data Stream as Disjoint Intervals Problem Description: Given a data stream input of non-negative integers a1, a2, ..., an, summarize the numbers seen so far as a list of disjoint intervals. Implement the SummaryRanges class:

SummaryRanges() Initializes the object with an empty stream. void addNum(int value) Adds the integer value to the stream. int[][] getIntervals() Returns a summary of the integers in the stream currently as a list of disjoint intervals [starti, endi]. The answer should be sorted by starti.

Example 1: Input ["SummaryRanges", "addNum", "getIntervals", "addNum", "getIntervals", "addNum", "getIntervals", "addNum", "getIntervals", "addNum", "getIntervals"] [[], [1], [], [3], [], [7], [], [2], [], [6], []] Output [null, null, [[1, 1]], null, [[1, 1], [3, 3]], null, [[1, 1], [3, 3], [7, 7]], null, [[1, 3], [7, 7]], null, [[1, 3], [6, 7]]]

Explanation SummaryRanges summaryRanges = new SummaryRanges(); summaryRanges.addNum(1); // arr = [1] summaryRanges.getIntervals(); // return [[1, 1]] summaryRanges.addNum(3); // arr = [1, 3] summaryRanges.getIntervals(); // return [[1, 1], [3, 3]] summaryRanges.addNum(7); // arr = [1, 3, 7] summaryRanges.getIntervals(); // return [[1, 1], [3, 3], [7, 7]] summaryRanges.addNum(2); // arr = [1, 2, 3, 7] summaryRanges.getIntervals(); // return [[1, 3], [7, 7]] summaryRanges.addNum(6); // arr = [1, 2, 3, 6, 7] summaryRanges.getIntervals(); // return [[1, 3], [6, 7]]

Constraints:

0 <= value <= 104 At most 3 * 104 calls will be made to addNum and getIntervals. At most 102 calls will be made to getIntervals.

Follow up: What if there are lots of merges and the number of disjoint intervals is small compared to the size of the data stream?

==
Problem Number: 1728 URL: https://leetcode.com/problems/russian-doll-envelopes Title: 354. Russian Doll Envelopes Problem Description: You are given a 2D array of integers envelopes where envelopes[i] = [wi, hi] represents the width and the height of an envelope. One envelope can fit into another if and only if both the width and height of one envelope are greater than the other envelope's width and height. Return the maximum number of envelopes you can Russian doll (i.e., put one inside the other). Note: You cannot rotate an envelope. Example 1: Input: envelopes = [[5,4],[6,4],[6,7],[2,3]] Output:

3 Explanation: The maximum number of envelopes you can Russian doll is 3 ([2,3] => [5,4] => [6,7]).

Example 2: Input: envelopes = [[1,1],[1,1],[1,1]] Output: 1

Constraints:

1 <= envelopes.length <= 105 envelopes[i].length == 2 1 <= wi, hi <= 105

==

Problem Number: 1729 URL: https://leetcode.com/problems/max-sum-of-rectangle-no-larger-than-k Title: 363. Max Sum of Rectangle No Larger Than K Problem Description: Given an m x n matrix matrix and an integer k, return the max sum of a rectangle in the matrix such that its sum is no larger than k. It is guaranteed that there will be a rectangle with a sum no larger than k. Example 1:

Input: matrix = [[1,0,1],[0,-2,3]], k = 2 Output: 2 Explanation: Because the sum of the blue rectangle [[0, 1], [-2, 3]] is 2, and 2 is the max number no larger than k (k = 2).

Example 2: Input: matrix = [[2,2,-1]], k = 3 Output: 3

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 100 -100 <= matrix[i][j] <= 100 -105 <= k <= 105

Follow up: What if the number of rows is much larger than the number of columns?

==

Problem Number: 1730 URL: https://leetcode.com/problems/insert-delete-getrandom-o1-duplicates-allowed Title: 381. Insert Delete GetRandom O(1) - Duplicates allowed Problem Description: RandomizedCollection is a data structure that contains a collection of numbers, possibly duplicates (i.e., a multiset). It should support inserting and removing specific elements and also reporting a random element. Implement the RandomizedCollection class:

RandomizedCollection() Initializes the empty RandomizedCollection object. bool insert(int val) Inserts an item val into the multiset, even if the item is already present. Returns true if the item is not present, false otherwise. bool remove(int val) Removes an item val from the multiset if present. Returns true if the item is present, false otherwise. Note that if val has multiple occurrences in the multiset, we only remove one of them. int getRandom() Returns a random element from the current multiset of elements. The probability of each element being returned is linearly related to the number of the same values the multiset contains.

You must implement the functions of the class such that each function works on average O(1) time complexity. Note: The test cases are generated such that getRandom will only be called if there is at least one item in the RandomizedCollection. Example 1: Input ["RandomizedCollection", "insert", "insert", "insert", "getRandom", "remove", "getRandom"] [[], [1], [1], [2], [], [1], []] Output [null, true, false, true, 2, true, 1]

Explanation RandomizedCollection randomizedCollection = new RandomizedCollection(); randomizedCollection.insert(1); // return true since the collection does not contain 1. // Inserts 1 into the collection. randomizedCollection.insert(1); // return false since the collection contains 1. // Inserts another 1 into the collection. Collection now contains [1,1]. randomizedCollection.insert(2); // return true since the collection does not contain 2. // Inserts 2 into the collection. Collection now contains [1,1,2]. randomizedCollection.getRandom(); // getRandom should: // - return 1 with probability 2/3, or // - return 2 with probability 1/3. randomizedCollection.remove(1); // return true since the collection contains 1. // Removes 1 from the collection. Collection now contains [1,2]. randomizedCollection.getRandom(); // getRandom should return 1 or 2, both equally likely.

Constraints:

-231 <= val <= 231 - 1 At most 2 * 105 calls in total will be made to insert, remove, and getRandom. There will be at least one element in the data structure when getRandom is called.

==

Problem Number: 1731 URL: https://leetcode.com/problems/perfect-rectangle Title: 391. Perfect Rectangle Problem Description: Given an array rectangles where rectangles[i] = [xi, yi, ai, bi] represents an axis-aligned rectangle. The bottom-left point of the rectangle is (xi, yi) and the top-right point of it is (ai, bi). Return true if all the rectangles together form an exact cover of a rectangular region. Example 1:

Input: rectangles = [[1,1,3,3],[3,1,4,2],[3,2,4,4],[1,3,2,4],[2,3,3,4]] Output: true Explanation: All 5 rectangles together form an exact cover of a rectangular region.

Example 2:

Input: rectangles = [[1,1,2,3],[1,3,2,4],[3,1,4,2],[3,2,4,4]] Output: false Explanation: Because there is a gap between the two rectangular regions.

Example 3:

Input: rectangles = [[1,1,3,3],[3,1,4,2],[1,3,2,4],[2,2,4,4]] Output: false Explanation: Because two of the rectangles overlap with each other.

Constraints:

1 <= rectangles.length <= 2 * 104 rectangles[i].length == 4 -105 <= xi, yi, ai, bi <= 105

== Problem Number: 1732 URL: https://leetcode.com/problems/frog-jump Title: 403. Frog Jump Problem Description: A frog is crossing a river. The river is divided into some number of units, and at each unit, there may or may not exist a stone. The frog can jump on a stone, but it must not jump into the water. Given a list of stones positions (in units) in sorted ascending order, determine if the frog can cross the river by landing on the last stone. Initially, the frog is on the first stone and assumes the first jump must be 1 unit. If the frog's last jump was k units, its next jump must be either k - 1, k, or k + 1 units. The frog can only jump in the forward direction. Example 1: Input: stones = [0,1,3,5,6,8,12,17] Output: true Explanation: The frog can jump to the last stone by jumping 1 unit to the 2nd stone, then 2 units to the 3rd stone, then 2 units to the 4th stone, then 3 units to the 6th stone, 4 units to the 7th stone, and 5 units to the 8th stone.

Example 2: Input: stones = [0,1,2,3,4,8,9,11] Output: false Explanation: There is no way to jump to the last stone as the gap between the 5th and 6th stone is too large.

Constraints:

2 <= stones.length <= 2000 0 <= stones[i] <= 231 - 1 stones[0] == 0 stones is sorted in a strictly increasing order.

== Problem Number: 1733 URL: https://leetcode.com/problems/trapping-rain-water-ii Title: 407. Trapping Rain Water II Problem Description: Given an m x n integer matrix heightMap representing the height of each unit cell in a 2D elevation map, return the volume of water it can trap after raining. Example 1:

Input: heightMap = [[1,4,3,1,3,2],[3,2,1,3,2,4],[2,3,3,2,3,1]] Output: 4 Explanation: After the rain, water is trapped between the blocks. We have two small ponds 1 and 3 units trapped. The total volume of water trapped is 4.

Example 2:

Input: heightMap = [[3,3,3,3,3],[3,2,2,2,3],[3,2,1,2,3],[3,2,2,2,3],[3,3,3,3,3]] Output: 10

Constraints:

m == heightMap.length n == heightMap[i].length 1 <= m, n <= 200 0 <= heightMap[i][j] <= 2 * 104

== Problem Number: 1734 URL: https://leetcode.com/problems/split-array-largest-sum Title: 410. Split Array Largest Sum Problem Description: Given an integer array nums and an integer k, split nums into k non-empty subarrays such that the largest sum of any subarray is minimized. Return the minimized largest sum of the split. A subarray is a contiguous part of the array. Example

1: Input: nums = [7,2,5,10,8], k = 2 Output: 18 Explanation: There are four ways to split nums into two subarrays. The best way is to split it into [7,2,5] and [10,8], where the largest sum among the two subarrays is only 18.

Example 2: Input: nums = [1,2,3,4,5], k = 2 Output: 9 Explanation: There are four ways to split nums into two subarrays. The best way is to split it into [1,2,3] and [4,5], where the largest sum among the two subarrays is only 9.

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] <= 106 1 <= k <= min(50, nums.length)

==
Problem Number: 1735 URL: https://leetcode.com/problems/strong-password-checker Title: 420. Strong Password Checker Problem Description: A password is considered strong if the below conditions are all met:

It has at least 6 characters and at most 20 characters. It contains at least one lowercase letter, at least one uppercase letter, and at least one digit. It does not contain three repeating characters in a row (i.e., "Baaabb0" is weak, but "Baaba0" is strong).

Given a string password, return the minimum number of steps required to make password strong. if password is already strong, return 0. In one step, you can:

Insert one character to password, Delete one character from password, or Replace one character of password with another character.

Example 1: Input: password = "a" Output: 5 Example 2: Input: password = "aA1" Output: 3 Example 3: Input: password = "1337C0d3" Output: 0

Constraints:

1 <= password.length <= 50 password consists of letters, digits, dot '.' or exclamation mark '!'.

==
Problem Number: 1736 URL: https://leetcode.com/problems/all-oone-data-structure Title: 432. All O'one Data Structure Problem Description: Design a data structure to store the strings' count with the ability to return the strings with minimum and maximum counts. Implement the AllOne class:

AllOne() Initializes the object of the data structure. inc(String key) Increments the count of the string key by 1. If key does not exist in the data structure, insert it with count 1. dec(String key) Decrements the count of the string key by 1. If the count of key is 0 after the decrement, remove it from the data structure. It is guaranteed that key exists in the data structure before the decrement. getMaxKey() Returns one of the keys with the maximal count. If no element exists, return an empty string "". getMinKey() Returns one of the keys with the minimum count. If no element exists, return an empty string "".

Note that each function must run in O(1) average time complexity. Example 1: Input ["AllOne", "inc", "inc", "getMaxKey", "getMinKey", "inc", "getMaxKey", "getMinKey"] [[], ["hello"], ["hello"], [], [], ["leet"], [], []] Output [null, null, null, "hello", "hello", null, "hello", "leet"]

Explanation AllOne allOne = new AllOne(); allOne.inc("hello"); allOne.inc("hello"); allOne.getMaxKey(); // return "hello" allOne.getMinKey(); // return "hello" allOne.inc("leet"); allOne.getMaxKey(); // return "hello" allOne.getMinKey(); // return "leet"

Constraints:

1 <= key.length <= 10 key consists of lowercase English letters. It is guaranteed that for each call to dec, key is existing in the data structure. At most 5 * 104 calls will be made to inc, dec, getMaxKey, and getMinKey.

==
Problem Number: 1737 URL: https://leetcode.com/problems/k-th-smallest-in-lexicographical-order Title: 440. K-th Smallest in Lexicographical Order Problem Description: Given two integers n and k, return the kth lexicographically smallest integer in the range [1, n]. Example 1: Input: n = 13, k = 2 Output: 10 Explanation: The lexicographical order is [1, 10, 11, 12, 13, 2, 3, 4, 5, 6, 7, 8, 9], so the second smallest number is 10.

Example 2: Input: n = 1, k = 1 Output: 1

Constraints:

1 <= k <= n <= 109

==
Problem Number: 1738 URL: https://leetcode.com/problems/arithmetic-slices-ii-subsequence Title: 446. Arithmetic Slices II - Subsequence Problem Description: Given an integer array nums, return the number of all the arithmetic subsequences of nums. A sequence of numbers is called arithmetic if it consists of at least three elements and if the difference between any two consecutive elements is the same.

For example, [1, 3, 5, 7, 9], [7, 7, 7, 7], and [3, -1, -5, -9] are arithmetic sequences. For example, [1, 1, 2, 5, 7] is not an arithmetic sequence.

A subsequence of an array is a sequence that can be formed by removing some elements (possibly none) of the array.

For example, [2,5,10] is a subsequence of [1,2,1,2,4,1,5,10].

The test cases are generated so that the answer fits in 32-bit integer. Example 1: Input: nums = [2,4,6,8,10] Output: 7 Explanation: All arithmetic subsequence slices are: [2,4,6] [4,6,8] [6,8,10] [2,4,6,8] [4,6,8,10] [2,4,6,8,10] [2,6,10]

Example 2: Input: nums = [7,7,7,7,7] Output: 16 Explanation: Any subsequence of this array is arithmetic.

Constraints:

1 <= nums.length <= 1000 -231 <= nums[i] <= 231 - 1

==
Problem Number: 1739 URL: https://leetcode.com/problems/poor-pigs Title: 458. Poor Pigs Problem Description: There are buckets buckets of liquid, where exactly one of the buckets is poisonous. To figure out which one is poisonous, you feed some number of (poor) pigs the liquid to see whether they will die or not. Unfortunately, you only have minutesToTest minutes to determine which bucket is poisonous. You can feed the pigs according to these steps:

Choose some live pigs to feed. For each pig, choose which buckets to feed it. The pig will consume all the chosen buckets simultaneously and will take no time. Each pig can feed from any number of buckets, and each bucket can be fed from by any number of pigs. Wait for minutesToDie minutes. You may not feed any other pigs during this time. After minutesToDie minutes have passed, any pigs that have been fed the poisonous bucket will die, and all others will survive. Repeat this process until you run out of time.

Given buckets, minutesToDie, and minutesToTest, return the minimum number of pigs needed to figure out which bucket is poisonous within the allotted time. Example 1: Input: buckets = 4, minutesToDie = 15, minutesToTest = 15 Output: 2 Explanation: We can determine the poisonous bucket as follows: At time 0, feed the first pig buckets 1 and 2, and feed the second pig buckets 2 and 3. At time 15, there are 4 possible outcomes: - If only the first pig dies, then bucket 1 must be poisonous. - If only the second pig dies, then bucket 3 must be poisonous. - If both pigs die, then bucket 2 must be poisonous. - If neither pig dies, then bucket 4 must be poisonous.

Example 2: Input: buckets = 4, minutesToDie = 15, minutesToTest = 30 Output: 2 Explanation: We can determine the poisonous bucket as follows: At time 0, feed the first pig bucket 1, and feed the second pig bucket 2. At time 15, there are 2 possible outcomes: - If either pig dies, then the poisonous bucket is the one it was fed. - If neither pig dies, then feed the first pig bucket 3, and feed the second pig bucket 4. At time 30, one of the two pigs must die, and the poisonous bucket is the one it was fed.

Constraints:

1 <= buckets <= 1000 1 <= minutesToDie <= minutesToTest <= 100

==
Problem Number: 1740 URL: https://leetcode.com/problems/lfu-cache Title: 460. LFU Cache Problem Description: Design and implement a data structure for a Least Frequently Used (LFU) cache. Implement the LFUCache class:

LFUCache(int capacity) Initializes the object with the capacity of the data structure. int get(int key) Gets the value of the key if the key exists in the cache. Otherwise, returns -1. void put(int key, int value) Update the value of the key if present, or inserts the key if not already present. When the cache reaches its capacity, it should invalidate and remove the least frequently used key before inserting a new item. For this problem, when there is a tie (i.e., two or more keys with the same frequency), the least recently used key would be invalidated.

To determine the least frequently used key, a use counter is maintained for each key in the cache. The key with the smallest use counter is the least frequently used key. When a key is first inserted into the cache, its use counter is set to 1 (due to the put operation). The use counter for a key in the cache is incremented either a get or put operation is called on it. The functions get and put must each run in O(1) average time complexity. Example 1: Input ["LFUCache", "put", "put", "get", "put", "get", "get", "put", "get", "get", "get"] [[2], [1, 1], [2, 2], [1], [3, 3], [2], [3], [4, 4], [1], [3], [4]] Output [null, null, null, 1, null, -1, 3, null, -1, 3, 4]

Explanation // cnt(x) = the use counter for key x // cache=[] will show the last used order for tiebreakers (leftmost element is most recent) LFUCache lfu = new LFUCache(2); lfu.put(1, 1); // cache=[1,_], cnt(1)=1 lfu.put(2, 2); // cache=[2,1], cnt(2)=1, cnt(1)=1 lfu.get(1); // return 1 // cache=[1,2], cnt(2)=1, cnt(1)=2 lfu.put(3, 3); // 2 is the LFU key because cnt(2)=1 is the smallest, invalidate 2. // cache=[3,1], cnt(3)=1, cnt(1)=2 lfu.get(2); // return -1 (not found) lfu.get(3); // return 3 // cache=[3,1], cnt(3)=2, cnt(1)=2 lfu.put(4, 4); // Both 1 and 3 have the same cnt, but 1 is LRU, invalidate 1. // cache=[4,3], cnt(4)=1, cnt(3)=2 lfu.get(1); // return -1 (not found) lfu.get(3); // return 3 // cache=[3,4], cnt(4)=1, cnt(3)=3 lfu.get(4); // return 4 // cache=[4,3], cnt(4)=2, cnt(3)=3

Constraints:

1 <= capacity <= 104 0 <= key <= 105 0 <= value <= 109 At most 2 * 105 calls will be made to get and put.

==

Problem Number: 1741 URL: https://leetcode.com/problems/count-the-repetitions Title: 466. Count The Repetitions Problem Description: We define str = [s, n] as the string str which consists of the string s concatenated n times.

For example, str == ["abc", 3] =="abcabcabc".

We define that string s1 can be obtained from string s2 if we can remove some characters from s2 such that it becomes s1.

For example, s1 = "abc" can be obtained from s2 = "abdbec" based on our definition by removing the bolded underlined characters.

You are given two strings s1 and s2 and two integers n1 and n2. You have the two strings str1 = [s1, n1] and str2 = [s2, n2]. Return the maximum integer m such that str = [str2, m] can be obtained from str1. Example 1: Input: s1 =

"acb", n1 = 4, s2 = "ab", n2 = 2 Output: 2 Example 2: Input: s1 = "acb", n1 = 1, s2 = "acb", n2 = 1 Output: 1

Constraints:

1 <= s1.length, s2.length <= 100 s1 and s2 consist of lowercase English letters. 1 <= n1, n2 <= 106

==

Problem Number: 1742 URL: https://leetcode.com/problems/concatenated-words Title: 472. Concatenated Words Problem Description: Given an array of strings words (without duplicates), return all the concatenated words in the given list of words. A concatenated word is defined as a string that is comprised entirely of at least two shorter words (not necesssarily distinct) in the given array. Example 1: Input: words = ["cat","cats","catsdogcats","dog","dogcatsdog","hippopotamuses","rat","ratcatdogcat"] Output: ["catsdogcats","dogcatsdog","ratcatdogcat"] Explanation: "catsdogcats" can be concatenated by "cats", "dog" and "cats"; "dogcatsdog" can be concatenated by "dog", "cats" and "dog"; "ratcatdogcat" can be concatenated by "rat", "cat", "dog" and "cat". Example 2: Input: words = ["cat","dog","catdog"] Output: ["catdog"]

Constraints:

1 <= words.length <= 104 1 <= words[i].length <= 30 words[i] consists of only lowercase English letters. All the strings of words are unique. 1 <= sum(words[i].length) <= 105

==

Problem Number: 1743 URL: https://leetcode.com/problems/largest-palindrome-product Title: 479. Largest Palindrome Product Problem Description: Given an integer n, return the largest palindromic integer that can be represented as the product of two n-digits integers. Since the answer can be very large, return it modulo 1337. Example 1: Input: n = 2 Output: 987 Explanation: 99 x 91 = 9009, 9009 % 1337 = 987

Example 2: Input: n = 1 Output: 9

Constraints:

1 <= n <= 8

==

Problem Number: 1744 URL: https://leetcode.com/problems/sliding-window-median Title: 480. Sliding Window Median Problem Description: The median is the middle value in an ordered integer list. If the size of the list is even, there is no middle value. So the median is the mean of the two middle values.

For examples, if arr = [2,3,4], the median is 3. For examples, if arr = [1,2,3,4], the median is (2 + 3) / 2 = 2.5.

You are given an integer array nums and an integer k. There is a sliding window of size k which is moving from the very left of the array to the very right. You can only see the k numbers in the window. Each time the sliding window moves right by one position. Return the median array for each window in the original array. Answers within 10-5 of the actual value will be accepted. Example 1: Input: nums = [1,3,-1,-3,5,3,6,7], k = 3 Output: [1.00000,-1.00000,-1.00000,3.00000,5.00000,6.00000] Explanation: Window position Median --------------- ----- [1 3 -1] -3 5 3 6 7 1 1 [3 -1 -3] 5 3 6 7 -1 1 3 [-1 -3 5] 3 6 7 -1 1 3 -1 [-3 5 3] 6 7 3 1 3 -1 -3 [5 3 6] 7 5 1 3 -1 -3 5 [3 6 7] 6

Example 2: Input: nums = [1,2,3,4,2,3,1,4,2], k = 3 Output: [2.00000,3.00000,3.00000,3.00000,2.00000,3.00000,2

Constraints:

1 <= k <= nums.length <= 105 -231 <= nums[i] <= 231 - 1

==

Problem Number: 1745 URL: https://leetcode.com/problems/smallest-good-base Title: 483. Smallest Good Base Problem Description: Given an integer n represented as a string, return the smallest good base of n. We call k >= 2 a good base of n, if all digits of n base k are 1's. Example 1: Input: n = "13" Output: "3" Explanation: 13 base 3 is 111.

Example 2: Input: n = "4681" Output: "8" Explanation: 4681 base 8 is 11111.

Example 3: Input: n = "1000000000000000000" Output: "999999999999999999" Explanation: 1000000000000000000 base 999999999999999999 is 11.

Constraints:

n is an integer in the range [3, 1018]. n does not contain any leading zeros.

==

Problem Number: 1746 URL: https://leetcode.com/problems/zuma-game Title: 488. Zuma Game Problem Description: You are playing a variation of the game Zuma. In this variation of Zuma, there is a single row of colored balls on a board, where each ball can be colored red 'R', yellow 'Y', blue 'B', green 'G', or white 'W'. You also have several colored balls in your hand. Your goal is to clear all of the balls from the board. On each turn:

Pick any ball from your hand and insert it in between two balls in the row or on either end of the row. If there is a group of three or more consecutive balls of the same color, remove the group of balls from the board.

If this removal causes more groups of three or more of the same color to form, then continue removing each group until there are none left.

If there are no more balls on the board, then you win the game. Repeat this process until you either win or do not have any more balls in your hand.

Given a string board, representing the row of balls on the board, and a string hand, representing the balls in your hand, return the minimum number of balls you have to insert to clear all the balls from the board. If you cannot clear all the balls from the board using the balls in your hand, return -1. Example 1: Input: board = "WRRBBW", hand = "RB" Output: -1 Explanation: It is impossible to clear all the balls. The best you can do is: - Insert 'R' so the board becomes WRRRBBW. WRRRBBW -> WBBW. - Insert 'B' so the board becomes WBBBW. WBBBW -> WW. There are still balls remaining on the board, and you are out of balls to insert. Example 2: Input: board = "WWR-RBBWW", hand = "WRBRW" Output: 2 Explanation: To make the board empty: - Insert 'R' so the board becomes WWRRRBBWW. WWRRRBBWW -> WWBBWW. - Insert 'B' so the board becomes WWBBBWW. WWBBBWW -> WWWW -> empty. 2 balls from your hand were needed to clear the board.

Example 3: Input: board = "G", hand = "GGGGG" Output: 2 Explanation: To make the board empty: - Insert 'G' so the board becomes GG. - Insert 'G' so the board becomes GGG. GGG -> empty. 2 balls from your hand were needed to clear the board.

Constraints:

1 <= board.length <= 16 1 <= hand.length <= 5 board and hand consist of the characters 'R', 'Y', 'B', 'G', and 'W'. The initial row of balls on the board will not have any groups of three or more consecutive balls of the same color.

==

Problem Number: 1747 URL: https://leetcode.com/problems/reverse-pairs Title: 493. Reverse Pairs Problem Description: Given an integer array nums, return the number of reverse pairs in the array. A reverse pair is a pair (i, j) where:

0 <= i < j < nums.length and nums[i] > 2 * nums[j].

Example 1: Input: nums = [1,3,2,3,1] Output: 2 Explanation: The reverse pairs are: (1, 4) --> nums[1] = 3, nums[4] = 1, 3 > 2 * 1 (3, 4) --> nums[3] = 3, nums[4] = 1, 3 > 2 * 1

Example 2: Input: nums = [2,4,3,5,1] Output: 3 Explanation: The reverse pairs are: (1, 4) --> nums[1] = 4, nums[4] = 1, 4 > 2 * 1 (2, 4) --> nums[2] = 3, nums[4] = 1, 3 > 2 * 1 (3, 4) --> nums[3] = 5, nums[4] = 1, 5 > 2 * 1

Constraints:

1 <= nums.length <= 5 * 104 -231 <= nums[i] <= 231 - 1

==

Problem Number: 1748 URL: https://leetcode.com/problems/ipo Title: 502. IPO Problem Description: Suppose LeetCode will start its IPO soon. In order to sell a good price of its shares to Venture Capital, LeetCode would like to work on some projects to increase its capital before the IPO. Since it has limited resources, it can only finish at most k distinct projects before the IPO. Help LeetCode design the best way to maximize its total capital after finishing at most k distinct projects. You are given n projects where the ith project has a pure profit profits[i] and a minimum capital of capital[i] is needed to start it. Initially, you have w capital. When you finish a project, you will obtain its pure profit and the profit will be added to your total capital. Pick a list of at most k distinct projects from given projects to maximize your final capital, and return the final maximized capital. The answer is guaranteed to fit in a 32-bit signed integer. Example 1: Input: k = 2, w = 0, profits = [1,2,3], capital = [0,1,1] Output: 4 Explanation: Since your initial capital is 0, you can only start the project indexed 0. After finishing it you will obtain profit 1 and your capital becomes 1. With capital 1, you can either start the project indexed 1 or the project indexed 2. Since you can choose at most 2 projects, you need to finish the project indexed 2 to get the maximum capital. Therefore, output the final maximized capital, which is 0 + 1 + 3 = 4.

Example 2: Input: k = 3, w = 0, profits = [1,2,3], capital = [0,1,2] Output: 6

Constraints:

1 <= k <= 105 0 <= w <= 109 n == profits.length n == capital.length 1 <= n <= 105 0 <= profits[i] <= 104 0 <= capital[i] <= 109

==
Problem Number: 1749 URL: https://leetcode.com/problems/freedom-trail Title: 514. Freedom Trail Problem Description: In the video game Fallout 4, the quest "Road to Freedom" requires players to reach a metal dial called the "Freedom Trail Ring" and use the dial to spell a specific keyword to open the door. Given a string ring that represents the code engraved on the outer ring and another string key that represents the keyword that needs to be spelled, return the minimum number of steps to spell all the characters in the keyword. Initially, the first character of the ring is aligned at the "12:00" direction. You should spell all the characters in key one by one by rotating ring clockwise or anticlockwise to make each character of the string key aligned at the "12:00" direction and then by pressing the center button. At the stage of rotating the ring to spell the key character key[i]:

You can rotate the ring clockwise or anticlockwise by one place, which counts as one step. The final purpose of the rotation is to align one of ring's characters at the "12:00" direction, where this character must equal key[i]. If the character key[i] has been aligned at the "12:00" direction, press the center button to spell, which also counts as one step. After the pressing, you could begin to spell the next character in the key (next stage). Otherwise, you have finished all the spelling.

Example 1:

Input: ring = "godding", key = "gd" Output: 4 Explanation: For the first key character 'g', since it is already in place, we just need 1 step to spell this character. For the second key character 'd', we need to rotate the ring "godding" anticlockwise by two steps to make it become "ddinggo". Also, we need 1 more step for spelling. So the final output is 4.

Example 2: Input: ring = "godding", key = "godding" Output: 13

Constraints:

1 <= ring.length, key.length <= 100 ring and key consist of only lower case English letters. It is guaranteed that key could always be spelled by rotating ring.

==
Problem Number: 1750 URL: https://leetcode.com/problems/super-washing-machines Title: 517. Super Washing Machines Problem Description: You have n super washing machines on a line. Initially, each washing machine has some dresses or is empty. For each move, you could choose any m (1 <= m <= n) washing machines, and pass one dress of each washing machine to one of its adjacent washing machines at the same time. Given an integer array machines representing the number of dresses in each washing machine from left to right on the line, return the minimum number of moves to make all the washing machines have the same number of dresses. If it is not possible to do it, return -1. Example 1: Input: machines = [1,0,5] Output: 3 Explanation: 1st move: 1 0 <-- 5 => 1 1 4 2nd move: 1 <-- 1 <-- 4 => 2 1 3 3rd move: 2 1 <-- 3 => 2 2 2

Example 2: Input: machines = [0,3,0] Output: 2 Explanation: 1st move: 0 <-- 3 0 => 1 2 0 2nd move: 1 2 --> 0 => 1 1 1

Example 3: Input: machines = [0,2,0] Output: -1 Explanation: It's impossible to make all three washing machines have the same number of dresses.

Constraints:

n == machines.length 1 <= n <= 104 0 <= machines[i] <= 105

==
Problem Number: 1751 URL: https://leetcode.com/problems/remove-boxes Title: 546. Remove Boxes Problem Description: You are given several boxes with different colors represented by different positive numbers. You may experience several rounds to remove boxes until there is no box left. Each time you can choose some continuous boxes with the same color (i.e., composed of k boxes, k >= 1), remove them and get k * k points. Return the maximum points you can get. Example 1: Input: boxes = [1,3,2,2,2,3,4,3,1] Output: 23 Explanation: [1, 3, 2, 2, 2, 3, 4, 3, 1] ----> [1, 3, 3, 4, 3, 1] (3*3=9 points) ----> [1, 3, 3, 3, 1] (1*1=1 points) ----> [1, 1] (3*3=9 points) ----> [] (2*2=4 points)

Example 2: Input: boxes = [1,1,1] Output: 9

Example 3: Input: boxes = [1] Output: 1

Constraints:

1 <= boxes.length <= 100 1 <= boxes[i] <= 100

==
Problem Number: 1752 URL: https://leetcode.com/problems/student-attendance-record-ii Title: 552. Student Attendance Record II Problem Description: An attendance record for a student can be represented as a string where each character signifies whether the student was absent, late, or present on that day. The record only contains the following three characters:

'A': Absent. 'L': Late. 'P': Present.

Any student is eligible for an attendance award if they meet both of the following criteria:

The student was absent ('A') for strictly fewer than 2 days total. The student was never late ('L') for 3 or more consecutive days.

Given an integer n, return the number of possible attendance records of length n that make a student eligible for an attendance award. The answer may be very large, so return it modulo 109 + 7. Example 1: Input: n = 2 Output: 8 Explanation: There are 8 records with length 2 that are eligible for an award: "PP", "AP", "PA", "LP", "PL", "AL", "LA", "LL" Only "AA" is not eligible because there are 2 absences (there need to be fewer than 2).

Example 2: Input: n = 1 Output: 3

Example 3: Input: n = 10101 Output: 183236316

Constraints:

1 <= n <= 105

==
Problem Number: 1753 URL: https://leetcode.com/problems/find-the-closest-palindrome Title: 564. Find the Closest Palindrome Problem Description: Given a string n representing an integer, return the closest integer (not including itself), which is a palindrome. If there is a tie, return the smaller one. The closest is defined as the absolute difference minimized between two integers. Example 1: Input: n = "123" Output: "121"

Example 2: Input: n = "1" Output: "0" Explanation: 0 and 2 are the closest palindromes but we return the smallest which is 0.

Constraints:

1 <= n.length <= 18 n consists of only digits. n does not have leading zeros. n is representing an integer in the range [1, 1018 - 1].

==
Problem Number: 1754 URL: https://leetcode.com/problems/erect-the-fence Title: 587. Erect the Fence Problem Description: You are given an array trees where trees[i] = [xi, yi] represents the location of a tree in the garden. Fence the entire garden using the minimum length of rope, as it is expensive. The garden is well-fenced only if all the trees are enclosed. Return the coordinates

of trees that are exactly located on the fence perimeter. You may return the answer in any order. Example 1:

Input: trees = [[1,1],[2,2],[2,0],[2,4],[3,3],[4,2]] Output: [[1,1],[2,0],[4,2],[3,3],[2,4]] Explanation: All the trees will be on the perimeter of the fence except the tree at [2, 2], which will be inside the fence.

Example 2:

Input: trees = [[1,2],[2,2],[4,2]] Output: [[4,2],[2,2],[1,2]] Explanation: The fence forms a line that passes through all the trees.

Constraints:

1 <= trees.length <= 3000 trees[i].length == 2 0 <= xi, yi <= 100 All the given positions are unique.

==

Problem Number: 1755 URL: https://leetcode.com/problems/tag-validator Title: 591. Tag Validator Problem Description: Given a string representing a code snippet, implement a tag validator to parse the code and return whether it is valid. A code snippet is valid if all the following rules hold:

The code must be wrapped in a valid closed tag. Otherwise, the code is invalid. A closed tag (not necessarily valid) has exactly the following format : <TAG_NAME>TAG_CONTENT</TAG_NAME>. Among them, <TAG_NAME> is the start tag, and </TAG_NAME> is the end tag. The TAG_NAME in start and end tags should be the same. A closed tag is valid if and only if the TAG_NAME and TAG_CONTENT are valid. A valid TAG_NAME only contain upper-case letters, and has length in range [1,9]. Otherwise, the TAG_NAME is invalid. A valid TAG_CONTENT may contain other valid closed tags, cdata and any characters (see note1) EXCEPT unmatched <, unmatched start and end tag, and unmatched or closed tags with invalid TAG_NAME. Otherwise, the TAG_CONTENT is invalid. A start tag is unmatched if no end tag exists with the same TAG_NAME, and vice versa. However, you also need to consider the issue of unbalanced when tags are nested. A < is unmatched if you cannot find a subsequent >. And when you find a < or </, all the subsequent characters until the next > should be parsed as TAG_NAME (not necessarily valid). The cdata has the following format : <![CDATA[CDATA_CONTENT]]>. The range of CDATA_CONTENT is defined as the characters between <![CDATA[and the first subsequent]]>. CDATA_CONTENT may contain any characters. The function of cdata is to forbid the validator to parse CDATA_CONTENT, so even it has some characters that can be parsed as tag (no matter valid or invalid), you should treat it as regular characters.

Example 1: Input: code = "<DIV>This is the first line <![CDATA[<div>]]></DIV>" Output: true Explanation: The code is wrapped in a closed tag : <DIV> and </DIV>. The TAG_NAME is valid, the TAG_CONTENT consists of some characters and cdata. Although CDATA_CONTENT has an unmatched start tag with invalid TAG_NAME, it should be considered as plain text, not parsed as a tag. So TAG_CONTENT is valid, and then the code is valid. Thus return true.

Example 2: Input: code = "<DIV»> ![cdata[]] <![CDATA[<div>]>]]>]]»]</DIV>" Output: true Explanation: We first separate the code into : start_tag|tag_content|end_tag. start_tag -> "<DIV>" end_tag -> "</DIV>" tag_content could also be separated into : text1|cdata|text2. text1 -> "» ![cdata[]] " cdata -> "<![CDATA[<div>]>]]>", where the CDATA_CONTENT is "<div>]>" text2 -> "]]»]" The reason why start_tag is NOT "<DIV»>" is because of the rule 6. The reason why cdata is NOT "<![CDATA[<div>]>]]>]]>" is because of the rule 7.

Example 3: Input: code = "<A> <B> </A> </B>" Output: false Explanation: Unbalanced. If "<A>" is closed, then "<B>" must be unmatched, and vice versa.

Constraints:

1 <= code.length <= 500 code consists of English letters, digits, '<', '>', '/', '!', '[', ']', '.', and ' '.

==
Problem Number: 1756 URL: https://leetcode.com/problems/non-negative-integers-without-consecutive-ones Title: 600. Non-negative Integers without Consecutive Ones Problem Description: Given a positive integer n, return the number of the integers in the range [0, n] whose binary representations do not contain consecutive ones. Example 1: Input: n = 5 Output: 5 Explanation: Here are the non-negative integers <= 5 with their corresponding binary representations: 0 : 0 1 : 1 2 : 10 3 : 11 4 : 100 5 : 101 Among them, only integer 3 disobeys the rule (two consecutive ones) and the other 5 satisfy the rule.

Example 2: Input: n = 1 Output: 2

Example 3: Input: n = 2 Output: 3

Constraints:

1 <= n <= 109

==
Problem Number: 1757 URL: https://leetcode.com/problems/k-inverse-pairs-array Title: 629. K Inverse Pairs Array Problem Description: For an integer array nums, an inverse pair is a pair of integers [i, j] where 0 <= i < j < nums.length and nums[i] > nums[j]. Given two integers n and k, return the number of different arrays consist of numbers from 1 to n such that there are exactly k inverse pairs. Since the answer can be huge, return it modulo 109 + 7. Example 1: Input: n = 3, k = 0 Output: 1 Explanation: Only the array [1,2,3] which consists of numbers from 1 to 3 has exactly 0 inverse pairs.

Example 2: Input: n = 3, k = 1 Output: 2 Explanation: The array [1,3,2] and [2,1,3] have exactly 1 inverse pair.

Constraints:

1 <= n <= 1000 0 <= k <= 1000

==
Problem Number: 1758 URL: https://leetcode.com/problems/course-schedule-iii Title: 630. Course Schedule III Problem Description: There are n different online courses numbered from 1 to n. You are given an array courses where courses[i] = [durationi, lastDayi] indicate that the ith course should be taken continuously for durationi days and must be finished before or on lastDayi. You will start on the 1st day and you cannot take two or more courses simultaneously. Return the maximum number of courses that you can take. Example 1: Input: courses = [[100,200],[200,1300],[1000,1250],[2000,3200]] Output: 3 Explanation: There are totally 4 courses, but you can take 3 courses at most: First, take the 1st course, it costs 100 days so you will finish it on the 100th day, and ready to take the next course on the 101st day. Second, take the 3rd course, it costs 1000 days so you will finish it on the 1100th day, and ready to take the next course on the 1101st day. Third, take the 2nd course, it costs 200 days so you will finish it on the 1300th day. The 4th course cannot be taken now, since you will finish it on the 3300th day, which exceeds the closed date.

Example 2: Input: courses = [[1,2]] Output: 1

Example 3: Input: courses = [[3,2],[4,3]] Output: 0

Constraints:

1 <= courses.length <= 104 1 <= durationi, lastDayi <= 104

==
Problem Number: 1759 URL: https://leetcode.com/problems/smallest-range-covering-elements-from-k-lists Title: 632. Smallest Range Covering Elements from K Lists Problem Description: You have k lists of sorted integers in non-decreasing order. Find the smallest range that includes at least one number from each of the k lists. We define the range [a, b] is smaller than range [c, d] if b - a < d - c or a < c if b - a == d - c. Example 1: Input: nums = [[4,10,15,24,26],[0,9,12,20],[5,18,22,30]] Output: [20,24] Explanation: List 1: [4, 10, 15, 24,26], 24 is in range [20,24]. List 2: [0, 9, 12, 20], 20 is in range [20,24]. List 3: [5, 18, 22, 30], 22 is in range [20,24].

Example 2: Input: nums = [[1,2,3],[1,2,3],[1,2,3]] Output: [1,1]

Constraints:

nums.length == k 1 <= k <= 3500 1 <= nums[i].length <= 50 -105 <= nums[i][j] <= 105 nums[i] is sorted in non-decreasing order.

==
Problem Number: 1760 URL: https://leetcode.com/problems/decode-ways-ii Title: 639. Decode Ways II Problem Description: A message containing letters from A-Z can be encoded into numbers using the following mapping: 'A' -> "1" 'B' -> "2" ... 'Z' -> "26"

To decode an encoded message, all the digits must be grouped then mapped back into letters using the reverse of the mapping above (there may be multiple ways). For example, "11106" can be mapped into:

"AAJF" with the grouping (1 1 10 6) "KJF" with the grouping (11 10 6)

Note that the grouping (1 11 06) is invalid because "06" cannot be mapped into 'F' since "6" is different from "06". In addition to the mapping above, an encoded message may contain the '*' character, which can represent any digit from '1' to '9' ('0' is excluded). For example, the encoded message "1*" may represent any of the encoded messages "11", "12", "13", "14", "15", "16", "17", "18", or "19". Decoding "1*" is equivalent to decoding any of the encoded messages it can represent. Given a string s consisting of digits and '*' characters, return the number of ways to decode it. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: s = "*" Output: 9 Explanation: The encoded message can represent any of the encoded messages "1", "2", "3", "4", "5", "6", "7", "8", or "9". Each of these can be decoded to the strings "A", "B", "C", "D", "E", "F", "G", "H", and "I" respectively. Hence, there are a total of 9 ways to decode "*".

Example 2: Input: s = "1*" Output: 18 Explanation: The encoded message can represent any of the encoded messages "11", "12", "13", "14", "15", "16", "17", "18", or "19". Each of these encoded messages have 2 ways to be decoded (e.g. "11" can be decoded to "AA" or "K"). Hence, there are a total of 9 * 2 = 18 ways to decode "1*".

Example 3: Input: s = "2*" Output: 15 Explanation: The encoded message can represent any of the encoded messages "21", "22", "23", "24", "25", "26", "27", "28", or "29". "21", "22", "23", "24", "25", and "26" have 2 ways of being decoded, but "27", "28", and "29" only have 1 way. Hence, there are a total of (6 * 2) + (3 * 1) = 12 + 3 = 15 ways to decode "2*".

Constraints:

1 <= s.length <= 105 s[i] is a digit or '*'.

==
Problem Number: 1761 URL: https://leetcode.com/problems/strange-printer Title: 664. Strange Printer Problem Description: There is a strange printer with the following two special properties:

The printer can only print a sequence of the same character each time. At each turn, the printer can print new characters starting from and ending at any place and will cover the original existing characters.

Given a string s, return the minimum number of turns the printer needed to print it. Example 1: Input: s = "aaabbb" Output: 2 Explanation: Print "aaa" first and then print "bbb".

Example 2: Input: s = "aba" Output: 2 Explanation: Print "aaa" first and then print "b" from the second place of the string, which will cover the existing character 'a'.

Constraints:

1 <= s.length <= 100 s consists of lowercase English letters.

==

Problem Number: 1762 URL: https://leetcode.com/problems/kth-smallest-number-in-multiplication-table Title: 668. Kth Smallest Number in Multiplication Table Problem Description: Nearly everyone has used the Multiplication Table. The multiplication table of size m x n is an integer matrix mat where mat[i][j] == i * j (1-indexed). Given three integers m, n, and k, return the kth smallest element in the m x n multiplication table. Example 1:

Input: m = 3, n = 3, k = 5 Output: 3 Explanation: The 5th smallest number is 3.

Example 2:

Input: m = 2, n = 3, k = 6 Output: 6 Explanation: The 6th smallest number is 6.

Constraints:

1 <= m, n <= 3 * 104 1 <= k <= m * n

==

Problem Number: 1763 URL: https://leetcode.com/problems/cut-off-trees-for-golf-event Title: 675. Cut Off Trees for Golf Event Problem Description: You are asked to cut off all the trees in a forest for a golf event. The forest is represented as an m x n matrix. In this matrix:

0 means the cell cannot be walked through. 1 represents an empty cell that can be walked through. A number greater than 1 represents a tree in a cell that can be walked through, and this number is the tree's height.

In one step, you can walk in any of the four directions: north, east, south, and west. If you are standing in a cell with a tree, you can choose whether to cut it off. You must cut off the trees in order from shortest to tallest. When you cut off a tree, the value at its cell becomes 1 (an empty cell). Starting from the point (0, 0), return the minimum steps you need to walk to cut off all the trees. If you cannot cut off all the trees, return -1. Note: The input is generated such that no two trees have the same height, and there is at least one tree needs to be cut off. Example 1:

Input: forest = [[1,2,3],[0,0,4],[7,6,5]] Output: 6 Explanation: Following the path above allows you to cut off the trees from shortest to tallest in 6 steps.

Example 2:

Input: forest = [[1,2,3],[0,0,0],[7,6,5]] Output: -1 Explanation: The trees in the bottom row cannot be accessed as the middle row is blocked.

Example 3: Input: forest = [[2,3,4],[0,0,5],[8,7,6]] Output: 6 Explanation: You can follow the same path as Example 1 to cut off all the trees. Note that you can cut off the first tree at (0, 0) before making any steps.

Constraints:

m == forest.length n == forest[i].length 1 <= m, n <= 50 0 <= forest[i][j] <= 109 Heights of all trees are distinct.

==
Problem Number: 1764 URL: https://leetcode.com/problems/24-game Title: 679. 24 Game Problem Description: You are given an integer array cards of length 4. You have four cards, each containing a number in the range [1, 9]. You should arrange the numbers on these cards in a mathematical expression using the operators [’+’, ’-’, ’*’, ’/’] and the parentheses ’(’ and ’)’ to get the value 24. You are restricted with the following rules:

The division operator ’/’ represents real division, not integer division.

For example, 4 / (1 - 2 / 3) = 4 / (1 / 3) = 12.

Every operation done is between two numbers. In particular, we cannot use ’-’ as a unary operator.

For example, if cards = [1, 1, 1, 1], the expression ”-1 - 1 - 1 - 1” is not allowed.

You cannot concatenate numbers together

For example, if cards = [1, 2, 1, 2], the expression ”12 + 12” is not valid.

Return true if you can get such expression that evaluates to 24, and false otherwise. Example 1: Input: cards = [4,1,8,7] Output: true Explanation: (8-4) * (7-1) = 24

Example 2: Input: cards = [1,2,1,2] Output: false

Constraints:

cards.length == 4 1 <= cards[i] <= 9

==
Problem Number: 1765 URL: https://leetcode.com/problems/redundant-connection-ii Title: 685. Redundant Connection II Problem Description: In this problem, a rooted tree is a directed graph such that, there is exactly one node (the root) for which all other nodes are descendants of this node, plus every node has exactly one parent, except for the root node which has no

parents. The given input is a directed graph that started as a rooted tree with n nodes (with distinct values from 1 to n), with one additional directed edge added. The added edge has two different vertices chosen from 1 to n, and was not an edge that already existed. The resulting graph is given as a 2D-array of edges. Each element of edges is a pair [ui, vi] that represents a directed edge connecting nodes ui and vi, where ui is a parent of child vi. Return an edge that can be removed so that the resulting graph is a rooted tree of n nodes. If there are multiple answers, return the answer that occurs last in the given 2D-array.   Example 1:

Input: edges = [[1,2],[1,3],[2,3]] Output: [2,3]

Example 2:

Input: edges = [[1,2],[2,3],[3,4],[4,1],[1,5]] Output: [4,1]

Constraints:

n == edges.length 3 <= n <= 1000 edges[i].length == 2 1 <= ui, vi <= n ui != vi

==
Problem Number: 1766 URL: https://leetcode.com/problems/maximum-sum-of-3-non-overlapping-subarrays Title: 689. Maximum Sum of 3 Non-Overlapping Subarrays Problem Description: Given an integer array nums and an integer k, find three non-overlapping subarrays of length k with maximum sum and return them. Return the result as a list of indices representing the starting position of each interval (0-indexed). If there are multiple answers, return the lexicographically smallest one.   Example 1: Input: nums = [1,2,1,2,6,7,5,1], k = 2 Output: [0,3,5] Explanation: Subarrays [1, 2], [2, 6], [7, 5] correspond to the starting indices [0, 3, 5]. We could have also taken [2, 1], but an answer of [1, 3, 5] would be lexicographically larger.

Example 2: Input: nums = [1,2,1,2,1,2,1,2,1], k = 2 Output: [0,2,4]

Constraints:

1 <= nums.length <= 2 * 104 1 <= nums[i] < 216 1 <= k <= floor(nums.length / 3)

==
Problem Number: 1767 URL: https://leetcode.com/problems/stickers-to-spell-word Title: 691. Stickers to Spell Word Problem Description: We are given n different types of stickers. Each sticker has a lowercase English word on it. You would like to spell out the given string target by cutting individual letters from your collection of stickers and rearranging them. You can use each sticker more than once if you want, and you have infinite quantities of each sticker. Return the minimum number of stickers that you need to spell out target. If the task is impossible, return -1. Note: In all test cases, all words were chosen randomly from the 1000 most common US English words, and target was chosen as a concatenation of two random words.   Example 1: Input: stickers

= ["with","example","science"], target = "thehat" Output: 3 Explanation: We can use 2 "with" stickers, and 1 "example" sticker. After cutting and rearrange the letters of those stickers, we can form the target "thehat". Also, this is the minimum number of stickers necessary to form the target string.

Example 2: Input: stickers = ["notice","possible"], target = "basicbasic" Output: -1 Explanation: We cannot form the target "basicbasic" from cutting letters from the given stickers.

Constraints:

n == stickers.length 1 <= n <= 50 1 <= stickers[i].length <= 10 1 <= target.length <= 15 stickers[i] and target consist of lowercase English letters.

==

Problem Number: 1768 URL: https://leetcode.com/problems/falling-squares Title: 699. Falling Squares Problem Description: There are several squares being dropped onto the X-axis of a 2D plane. You are given a 2D integer array positions where positions[i] = [lefti, sideLengthi] represents the ith square with a side length of sideLengthi that is dropped with its left edge aligned with X-coordinate lefti. Each square is dropped one at a time from a height above any landed squares. It then falls downward (negative Y direction) until it either lands on the top side of another square or on the X-axis. A square brushing the left/right side of another square does not count as landing on it. Once it lands, it freezes in place and cannot be moved. After each square is dropped, you must record the height of the current tallest stack of squares. Return an integer array ans where ans[i] represents the height described above after dropping the ith square. Example 1:

Input: positions = [[1,2],[2,3],[6,1]] Output: [2,5,5] Explanation: After the first drop, the tallest stack is square 1 with a height of 2. After the second drop, the tallest stack is squares 1 and 2 with a height of 5. After the third drop, the tallest stack is still squares 1 and 2 with a height of 5. Thus, we return an answer of [2, 5, 5].

Example 2: Input: positions = [[100,100],[200,100]] Output: [100,100] Explanation: After the first drop, the tallest stack is square 1 with a height of 100. After the second drop, the tallest stack is either square 1 or square 2, both with heights of 100. Thus, we return an answer of [100, 100]. Note that square 2 only brushes the right side of square 1, which does not count as landing on it.

Constraints:

1 <= positions.length <= 1000 1 <= lefti <= 108 1 <= sideLengthi <= 106

==

Problem Number: 1769 URL: https://leetcode.com/problems/random-pick-with-blacklist Title: 710. Random Pick with Blacklist Problem Description: You are given an integer n and an array of unique integers blacklist. Design an algorithm to pick a random integer in the range [0, n - 1] that is not in blacklist. Any integer that is in the mentioned range and not in blacklist should be equally likely to be returned. Optimize your algorithm such that it minimizes the number of calls to the built-in random function of your language. Implement the Solution class:

Solution(int n, int[] blacklist) Initializes the object with the integer n and the blacklisted integers blacklist. int pick() Returns a random integer in the range [0, n - 1] and not in blacklist.

Example 1: Input ["Solution", "pick", "pick", "pick", "pick", "pick", "pick", "pick"] [[7, [2, 3, 5]], [], [], [], [], [], [], []] Output [null, 0, 4, 1, 6, 1, 0, 4]

Explanation Solution solution = new Solution(7, [2, 3, 5]); solution.pick(); // return 0, any integer from [0,1,4,6] should be ok. Note that for every call of pick, // 0, 1, 4, and 6 must be equally likely to be returned (i.e., with probability 1/4). solution.pick(); // return 4 solution.pick(); // return 1 solution.pick(); // return 6 solution.pick(); // return 1 solution.pick(); // return 0 solution.pick(); // return 4

Constraints:

1 <= n <= 109 0 <= blacklist.length <= min(105, n - 1) 0 <= blacklist[i] < n All the values of blacklist are unique. At most 2 * 104 calls will be made to pick.

==

Problem Number: 1770 URL: https://leetcode.com/problems/range-module Title: 715. Range Module Problem Description: A Range Module is a module that tracks ranges of numbers. Design a data structure to track the ranges represented as half-open intervals and query about them. A half-open interval [left, right) denotes all the real numbers x where left <= x < right. Implement the RangeModule class:

RangeModule() Initializes the object of the data structure. void addRange(int left, int right) Adds the half-open interval [left, right), tracking every real number in that interval. Adding an interval that partially overlaps with currently tracked numbers should add any numbers in the interval [left, right) that are not already tracked. boolean queryRange(int left, int right) Returns true if every real number in the interval [left, right) is currently being tracked, and false otherwise. void removeRange(int left, int right) Stops tracking every real number currently being tracked in the half-open interval [left, right).

Example 1: Input ["RangeModule", "addRange", "removeRange", "queryRange", "queryRange", "queryRange"] [[], [10, 20], [14, 16], [10, 14], [13, 15], [16, 17]] Output [null, null, null, true, false, true]

Explanation RangeModule rangeModule = new RangeModule(); rangeModule.addRange(10, 20); rangeModule.removeRange(14, 16); rangeModule.queryRange(10, 14); // return True,(Every number in [10, 14) is being tracked) rangeModule.queryRange(13, 15); // return False,(Numbers like 14, 14.03, 14.17 in [13, 15) are not being tracked) rangeModule.queryRange(16, 17); // return True, (The number 16 in [16, 17) is still being tracked, despite the remove operation)

Constraints:

1 <= left < right <= 109 At most 104 calls will be made to addRange, queryRange, and removeRange.

==
Problem Number: 1771 URL: https://leetcode.com/problems/find-k-th-smallest-pair-distance Title: 719. Find K-th Smallest Pair Distance Problem Description: The distance of a pair of integers a and b is defined as the absolute difference between a and b. Given an integer array nums and an integer k, return the kth smallest distance among all the pairs nums[i] and nums[j] where 0 <= i < j < nums.length. Example 1: Input: nums = [1,3,1], k = 1 Output: 0 Explanation: Here are all the pairs: (1,3) -> 2 (1,1) -> 0 (3,1) -> 2 Then the 1st smallest distance pair is (1,1), and its distance is 0.

Example 2: Input: nums = [1,1,1], k = 2 Output: 0

Example 3: Input: nums = [1,6,1], k = 3 Output: 5

Constraints:

n == nums.length 2 <= n <= 104 0 <= nums[i] <= 106 1 <= k <= n * (n - 1) / 2

==
Problem Number: 1772 URL: https://leetcode.com/problems/number-of-atoms Title: 726. Number of Atoms Problem Description: Given a string formula representing a chemical formula, return the count of each atom. The atomic element always starts with an uppercase character, then zero or more lowercase letters, representing the name. One or more digits representing that element's count may follow if the count is greater than 1. If the count is 1, no digits will follow.

For example, "H2O" and "H2O2" are possible, but "H1O2" is impossible.

Two formulas are concatenated together to produce another formula.

For example, "H2O2He3Mg4" is also a formula.

A formula placed in parentheses, and a count (optionally added) is also a formula.

For example, "(H2O2)" and "(H2O2)3" are formulas.

Return the count of all elements as a string in the following form: the first name (in sorted order), followed by its count (if that count is more than 1), followed by the second name (in sorted order), followed by its count (if that count is more than 1), and so on. The test cases are generated so that all the values in the output fit in a 32-bit integer. Example 1: Input: formula = "H2O" Output: "H2O" Explanation: The count of elements are {'H': 2, 'O': 1}.

Example 2: Input: formula = "Mg(OH)2" Output: "H2MgO2" Explanation: The count of elements are {'H': 2, 'Mg': 1, 'O': 2}.

Example 3: Input: formula = "K4(ON(SO3)2)2" Output: "K4N2O14S4" Explanation: The count of elements are {'K': 4, 'N': 2, 'O': 14, 'S': 4}.

Constraints:

1 <= formula.length <= 1000 formula consists of English letters, digits, '(', and ')'. formula is always valid.

==
Problem Number: 1773 URL: https://leetcode.com/problems/count-different-palindromic-subsequences Title: 730. Count Different Palindromic Subsequences Problem Description: Given a string s, return the number of different non-empty palindromic subsequences in s. Since the answer may be very large, return it modulo 109 + 7. A subsequence of a string is obtained by deleting zero or more characters from the string. A sequence is palindromic if it is equal to the sequence reversed. Two sequences a1, a2, ... and b1, b2, ... are different if there is some i for which ai != bi. Example 1: Input: s = "bccb" Output: 6 Explanation: The 6 different non-empty palindromic subsequences are 'b', 'c', 'bb', 'cc', 'bcb', 'bccb'. Note that 'bcb' is counted only once, even though it occurs twice.

Example 2: Input: s = "abcdabcdabcdabcdabcdabcdabcdabcddcbadcbadcbadcbadcbadcbadcbadcba" Output: 104860361 Explanation: There are 3104860382 different non-empty palindromic subsequences, which is 104860361 modulo 109 + 7.

Constraints:

1 <= s.length <= 1000 s[i] is either 'a', 'b', 'c', or 'd'.

==
Problem Number: 1774 URL: https://leetcode.com/problems/my-calendar-iii Title: 732. My Calendar III Problem Description: A k-booking happens when k events have some non-empty intersection (i.e., there is some time that is common to all k events.) You are given some events [startTime, endTime), after each given event, return an integer k representing the maximum k-booking between all the previous events. Implement the MyCalendarThree class:

MyCalendarThree() Initializes the object. int book(int startTime, int endTime) Returns an integer k representing the largest integer such that there exists a k-booking in the calendar.

Example 1: Input ["MyCalendarThree", "book", "book", "book", "book", "book", "book"] [[], [10, 20], [50, 60], [10, 40], [5, 15], [5, 10], [25, 55]] Output [null, 1, 1, 2, 3, 3, 3]

Explanation MyCalendarThree myCalendarThree = new MyCalendarThree(); myCalendarThree.book(10, 20); // return 1 myCalendarThree.book(50, 60); // return 1 myCalendarThree.book(10, 40); // return 2 myCalendarThree.book(5, 15); // return 3 myCalendarThree.book(5, 10); // return 3 myCalendarThree.book(25, 55); // return 3

Constraints:

0 <= startTime < endTime <= 109 At most 400 calls will be made to book.

==
Problem Number: 1775 URL: https://leetcode.com/problems/parse-lisp-expression Title: 736. Parse Lisp Expression Problem Description: You are given a string expression representing a Lisp-like expression to return the integer value of. The syntax for these expressions is given as follows.

An expression is either an integer, let expression, add expression, mult expression, or an assigned variable. Expressions always evaluate to a single integer. (An integer could be positive or negative.) A let expression takes the form "(let v1 e1 v2 e2 ... vn en expr)", where let is always the string "let", then there are one or more pairs of alternating variables and expressions, meaning that the first variable v1 is assigned the value of the expression e1, the second variable v2 is assigned the value of the expression e2, and so on sequentially; and then the value of this let expression is the value of the expression expr. An add expression takes the form "(add e1 e2)" where add is always the string "add", there are always two expressions e1, e2 and the result is the addition of the evaluation of e1 and the evaluation of e2. A mult expression takes the form "(mult e1 e2)" where mult is always the string "mult", there are always two expressions e1, e2 and the result is the multiplication of the evaluation of e1 and the evaluation of e2. For this question, we will use a smaller subset of variable names. A variable starts with a lowercase letter, then zero or more lowercase letters or digits. Additionally, for your convenience, the names "add", "let", and "mult" are protected and will never be used as variable names. Finally, there is the concept of scope. When an expression of a variable name is evaluated, within the context of that evaluation, the innermost scope (in terms of parentheses) is checked first for the value of that variable, and then outer scopes are checked sequentially. It is guaranteed that every expression is legal. Please see the examples for more details on the scope.

Example 1: Input: expression = "(let x 2 (mult x (let x 3 y 4 (add x y))))" Output: 14 Explanation: In the expression (add x y), when checking for the value of the variable x, we check from the innermost scope to the outermost in the context of the variable we are trying to evaluate. Since x = 3 is found first, the value of x is 3.

Example 2: Input: expression = "(let x 3 x 2 x)" Output: 2 Explanation: Assignment in let statements is processed sequentially.

Example 3: Input: expression = "(let x 1 y 2 x (add x y) (add x y))" Output:

5 Explanation: The first (add x y) evaluates as 3, and is assigned to x. The second (add x y) evaluates as 3+2 = 5.

Constraints:

1 <= expression.length <= 2000 There are no leading or trailing spaces in expression. All tokens are separated by a single space in expression. The answer and all intermediate calculations of that answer are guaranteed to fit in a 32-bit integer. The expression is guaranteed to be legal and evaluate to an integer.

==
Problem Number: 1776 URL: https://leetcode.com/problems/cherry-pickup Title: 741. Cherry Pickup Problem Description: You are given an n x n grid representing a field of cherries, each cell is one of three possible integers.

0 means the cell is empty, so you can pass through, 1 means the cell contains a cherry that you can pick up and pass through, or -1 means the cell contains a thorn that blocks your way.

Return the maximum number of cherries you can collect by following the rules below:

Starting at the position (0, 0) and reaching (n - 1, n - 1) by moving right or down through valid path cells (cells with value 0 or 1). After reaching (n - 1, n - 1), returning to (0, 0) by moving left or up through valid path cells. When passing through a path cell containing a cherry, you pick it up, and the cell becomes an empty cell 0. If there is no valid path between (0, 0) and (n - 1, n - 1), then no cherries can be collected.

Example 1:

Input: grid = [[0,1,-1],[1,0,-1],[1,1,1]] Output: 5 Explanation: The player started at (0, 0) and went down, down, right right to reach (2, 2). 4 cherries were picked up during this single trip, and the matrix becomes [[0,1,-1],[0,0,-1],[0,0,0]]. Then, the player went left, up, up, left to return home, picking up one more cherry. The total number of cherries picked up is 5, and this is the maximum possible.

Example 2: Input: grid = [[1,1,-1],[1,-1,1],[-1,1,1]] Output: 0

Constraints:

n == grid.length n == grid[i].length 1 <= n <= 50 grid[i][j] is -1, 0, or 1. grid[0][0] != -1 grid[n - 1][n - 1] != -1

==
Problem Number: 1777 URL: https://leetcode.com/problems/prefix-and-suffix-search Title: 745. Prefix and Suffix Search Problem Description: Design a special dictionary that searches the words in it by a prefix and a suffix. Implement the WordFilter class:

WordFilter(string[] words) Initializes the object with the words in the dictionary. f(string pref, string suff) Returns the index of the word in the dictionary, which has the prefix pref and the suffix suff. If there is more than one valid index, return the largest of them. If there is no such word in the dictionary, return -1.

Example 1: Input ["WordFilter", "f"] [[["apple"]], ["a", "e"]] Output [null, 0] Explanation WordFilter wordFilter = new WordFilter(["apple"]); wordFilter.f("a", "e"); // return 0, because the word at index 0 has prefix = "a" and suffix = "e".

Constraints:

1 <= words.length <= 104 1 <= words[i].length <= 7 1 <= pref.length, suff.length <= 7 words[i], pref and suff consist of lowercase English letters only. At most 104 calls will be made to the function f.

==
Problem Number: 1778 URL: https://leetcode.com/problems/contain-virus Title: 749. Contain Virus Problem Description: A virus is spreading rapidly, and your task is to quarantine the infected area by installing walls. The world is modeled as an m x n binary grid isInfected, where isInfected[i][j] == 0 represents uninfected cells, and isInfected[i][j] == 1 represents cells contaminated with the virus. A wall (and only one wall) can be installed between any two 4-directionally adjacent cells, on the shared boundary. Every night, the virus spreads to all neighboring cells in all four directions unless blocked by a wall. Resources are limited. Each day, you can install walls around only one region (i.e., the affected area (continuous block of infected cells) that threatens the most uninfected cells the following night). There will never be a tie. Return the number of walls used to quarantine all the infected regions. If the world will become fully infected, return the number of walls used. Example 1:

Input: isInfected = [[0,1,0,0,0,0,0,1],[0,1,0,0,0,0,0,1],[0,0,0,0,0,0,0,1],[0,0,0,0,0,0,0,0]] Output: 10 Explanation: There are 2 contaminated regions. On the first day, add 5 walls to quarantine the viral region on the left. The board after the virus spreads is:

On the second day, add 5 walls to quarantine the viral region on the right. The virus is fully contained.

Example 2:

Input: isInfected = [[1,1,1],[1,0,1],[1,1,1]] Output: 4 Explanation: Even though there is only one cell saved, there are 4 walls built. Notice that walls are only built on the shared boundary of two different cells.

Example 3: Input: isInfected = [[1,1,1,0,0,0,0,0,0],[1,0,1,0,1,1,1,1,1],[1,1,1,0,0,0,0,0,0]] Output: 13 Explanation: The region on the left only builds two new walls.

Constraints:

m == isInfected.length n == isInfected[i].length 1 <= m, n <= 50 isInfected[i][j] is either 0 or 1. There is always a contiguous viral region throughout the described process that will infect strictly more uncontaminated squares in the next round.

===

Problem Number: 1779 URL: https://leetcode.com/problems/cracking-the-safe Title: 753. Cracking the Safe Problem Description: There is a safe protected by a password. The password is a sequence of n digits where each digit can be in the range [0, k - 1]. The safe has a peculiar way of checking the password. When you enter in a sequence, it checks the most recent n digits that were entered each time you type a digit.

For example, the correct password is "345" and you enter in "012345":

After typing 0, the most recent 3 digits is "0", which is incorrect. After typing 1, the most recent 3 digits is "01", which is incorrect. After typing 2, the most recent 3 digits is "012", which is incorrect. After typing 3, the most recent 3 digits is "123", which is incorrect. After typing 4, the most recent 3 digits is "234", which is incorrect. After typing 5, the most recent 3 digits is "345", which is correct and the safe unlocks.

Return any string of minimum length that will unlock the safe at some point of entering it. Example 1: Input: n = 1, k = 2 Output: "10" Explanation: The password is a single digit, so enter each digit. "01" would also unlock the safe.

Example 2: Input: n = 2, k = 2 Output: "01100" Explanation: For each possible password: - "00" is typed in starting from the 4th digit. - "01" is typed in starting from the 1st digit. - "10" is typed in starting from the 3rd digit. - "11" is typed in starting from the 2nd digit. Thus "01100" will unlock the safe. "10011", and "11001" would also unlock the safe.

Constraints:

1 <= n <= 4 1 <= k <= 10 1 <= kn <= 4096

===

Problem Number: 1780 URL: https://leetcode.com/problems/set-intersection-size-at-least-two Title: 757. Set Intersection Size At Least Two Problem Description: You are given a 2D integer array intervals where intervals[i] = [starti, endi] represents all the integers from starti to endi inclusively. A containing set is an array nums where each interval from intervals has at least two integers in nums.

For example, if intervals = [[1,3], [3,7], [8,9]], then [1,2,4,7,8,9] and [2,3,4,8,9] are containing sets.

Return the minimum possible size of a containing set. Example 1: Input: intervals = [[1,3],[3,7],[8,9]] Output: 5 Explanation: let nums = [2, 3, 4, 8, 9]. It can be shown that there cannot be any containing array of size 4.

Example 2: Input: intervals = [[1,3],[1,4],[2,5],[3,5]] Output: 3 Explanation: let nums = [2, 3, 4]. It can be shown that there cannot be any containing array of size 2.

Example 3: Input: intervals = [[1,2],[2,3],[2,4],[4,5]] Output: 5 Explanation: let nums = [1, 2, 3, 4, 5]. It can be shown that there cannot be any containing array of size 4.

Constraints:

1 <= intervals.length <= 3000 intervals[i].length == 2 0 <= starti < endi <= 108

==

Problem Number: 1781 URL: https://leetcode.com/problems/special-binary-string Title: 761. Special Binary String Problem Description: Special binary strings are binary strings with the following two properties:

The number of 0's is equal to the number of 1's. Every prefix of the binary string has at least as many 1's as 0's.

You are given a special binary string s. A move consists of choosing two consecutive, non-empty, special substrings of s, and swapping them. Two strings are consecutive if the last character of the first string is exactly one index before the first character of the second string. Return the lexicographically largest resulting string possible after applying the mentioned operations on the string. Example 1: Input: s = ”11011000” Output: ”11100100” Explanation: The strings ”10” [occuring at s[1]] and ”1100” [at s[3]] are swapped. This is the lexicographically largest string possible after some number of swaps.

Example 2: Input: s = ”10” Output: ”10”

Constraints:

1 <= s.length <= 50 s[i] is either '0' or '1'. s is a special binary string.

==

Problem Number: 1782 URL: https://leetcode.com/problems/couples-holding-hands Title: 765. Couples Holding Hands Problem Description: There are n couples sitting in 2n seats arranged in a row and want to hold hands. The people and seats are represented by an integer array row where row[i] is the ID of the person sitting in the ith seat. The couples are numbered in order, the first couple being (0, 1), the second couple being (2, 3), and so on with the last couple being (2n - 2, 2n - 1). Return the minimum number of swaps so that every couple is sitting side by side. A swap consists of choosing any two people, then they stand up and switch seats. Example 1: Input: row = [0,2,1,3] Output: 1 Explanation: We only need to swap the second (row[1]) and third (row[2]) person.

Example 2: Input: row = [3,2,0,1] Output: 0 Explanation: All couples are already seated side by side.

Constraints:

2n == row.length 2 <= n <= 30 n is even. 0 <= row[i] < 2n All the elements of row are unique.

===
Problem Number: 1783 URL: https://leetcode.com/problems/max-chunks-to-make-sorted-ii Title: 768. Max Chunks To Make Sorted II Problem Description: You are given an integer array arr. We split arr into some number of chunks (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array. Return the largest number of chunks we can make to sort the array.   Example 1: Input: arr = [5,4,3,2,1] Output: 1 Explanation: Splitting into two or more chunks will not return the required result. For example, splitting into [5, 4], [3, 2, 1] will result in [4, 5, 1, 2, 3], which isn't sorted.

Example 2: Input: arr = [2,1,3,4,4] Output: 4 Explanation: We can split into two chunks, such as [2, 1], [3, 4, 4]. However, splitting into [2, 1], [3], [4], [4] is the highest number of chunks possible.

Constraints:

1 <= arr.length <= 2000 0 <= arr[i] <= 108

===
Problem Number: 1784 URL: https://leetcode.com/problems/basic-calculator-iv Title: 770. Basic Calculator IV Problem Description: Given an expression such as expression = "e + 8 - a + 5" and an evaluation map such as {"e": 1} (given in terms of evalvars = ["e"] and evalints = [1]), return a list of tokens representing the simplified expression, such as ["-1*a","14"]

An expression alternates chunks and symbols, with a space separating each chunk and symbol. A chunk is either an expression in parentheses, a variable, or a non-negative integer. A variable is a string of lowercase letters (not including digits.) Note that variables can be multiple letters, and note that variables never have a leading coefficient or unary operator like "2x" or "-x".

Expressions are evaluated in the usual order: brackets first, then multiplication, then addition and subtraction.

For example, expression = "1 + 2 * 3" has an answer of ["7"].

The format of the output is as follows:

For each term of free variables with a non-zero coefficient, we write the free variables within a term in sorted order lexicographically.

For example, we would never write a term like "b*a*c", only "a*b*c".

Terms have degrees equal to the number of free variables being multiplied, counting multiplicity. We write the largest degree terms of our answer first, breaking ties by lexicographic order ignoring the leading coefficient of the term.

For example, "a*a*b*c" has degree 4.

The leading coefficient of the term is placed directly to the left with an asterisk separating it from the variables (if they exist.) A leading coefficient of 1 is still printed. An example of a well-formatted answer is ["-2*a*a*a", "3*a*a*b", "3*b*b", "4*a", "5*c", "-6"]. Terms (including constant terms) with coefficient 0 are not included.

For example, an expression of "0" has an output of [].

Note: You may assume that the given expression is always valid. All intermediate results will be in the range of [-231, 231 - 1]. Example 1: Input: expression = "e + 8 - a + 5", evalvars = ["e"], evalints = [1] Output: ["-1*a","14"]

Example 2: Input: expression = "e - 8 + temperature - pressure", evalvars = ["e", "temperature"], evalints = [1, 12] Output: ["-1*pressure","5"]

Example 3: Input: expression = "(e + 8) * (e - 8)", evalvars = [], evalints = [] Output: ["1*e*e","-64"]

Constraints:

1 <= expression.length <= 250 expression consists of lowercase English letters, digits, '+', '-', '*', '(', ')', ' '. expression does not contain any leading or trailing spaces. All the tokens in expression are separated by a single space. 0 <= evalvars.length <= 100 1 <= evalvars[i].length <= 20 evalvars[i] consists of lowercase English letters. evalints.length == evalvars.length -100 <= evalints[i] <= 100

==

Problem Number: 1785 URL: https://leetcode.com/problems/sliding-puzzle Title: 773. Sliding Puzzle Problem Description: On an 2 x 3 board, there are five tiles labeled from 1 to 5, and an empty square represented by 0. A move consists of choosing 0 and a 4-directionally adjacent number and swapping it. The state of the board is solved if and only if the board is [[1,2,3],[4,5,0]]. Given the puzzle board board, return the least number of moves required so that the state of the board is solved. If it is impossible for the state of the board to be solved, return -1. Example 1:

Input: board = [[1,2,3],[4,0,5]] Output: 1 Explanation: Swap the 0 and the 5 in one move.

Example 2:

Input: board = [[1,2,3],[5,4,0]] Output: -1 Explanation: No number of moves will make the board solved.

Example 3:

Input: board = [[4,1,2],[5,0,3]] Output: 5 Explanation: 5 is the smallest number of moves that solves the board. An example path: After move 0: [[4,1,2],[5,0,3]] After move 1: [[4,1,2],[0,5,3]] After move 2: [[0,1,2],[4,5,3]] After move 3: [[1,0,2],[4,5,3]] After move 4: [[1,2,0],[4,5,3]] After move 5: [[1,2,3],[4,5,0]]

Constraints:

board.length == 2 board[i].length == 3 0 <= board[i][j] <= 5 Each value board[i][j] is unique.

==
Problem Number: 1786 URL: https://leetcode.com/problems/swim-in-rising-water Title: 778. Swim in Rising Water Problem Description: You are given an n x n integer matrix grid where each value grid[i][j] represents the elevation at that point (i, j). The rain starts to fall. At time t, the depth of the water everywhere is t. You can swim from a square to another 4-directionally adjacent square if and only if the elevation of both squares individually are at most t. You can swim infinite distances in zero time. Of course, you must stay within the boundaries of the grid during your swim. Return the least time until you can reach the bottom right square (n - 1, n - 1) if you start at the top left square (0, 0). Example 1:

Input: grid = [[0,2],[1,3]] Output: 3 Explanation: At time 0, you are in grid location (0, 0). You cannot go anywhere else because 4-directionally adjacent neighbors have a higher elevation than t = 0. You cannot reach point (1, 1) until time 3. When the depth of water is 3, we can swim anywhere inside the grid.

Example 2:

Input: grid = [[0,1,2,3,4],[24,23,22,21,5],[12,13,14,15,16],[11,17,18,19,20],[10,9,8,7,6]] Output: 16 Explanation: The final route is shown. We need to wait until time 16 so that (0, 0) and (4, 4) are connected.

Constraints:

n == grid.length n == grid[i].length 1 <= n <= 50 0 <= grid[i][j] < n2 Each value grid[i][j] is unique.

==
Problem Number: 1787 URL: https://leetcode.com/problems/reaching-points Title: 780. Reaching Points Problem Description: Given four integers sx, sy, tx, and ty, return true if it is possible to convert the point (sx, sy) to the point (tx, ty) through some operations, or false otherwise. The allowed operation on some point (x, y) is to convert it to either (x, x + y) or (x + y, y). Example 1: Input: sx = 1, sy = 1, tx = 3, ty = 5 Output: true Explanation: One series of moves that transforms the starting point to the target is: (1, 1) -> (1, 2) (1, 2) -> (3, 2) (3, 2) -> (3, 5)

Example 2: Input: sx = 1, sy = 1, tx = 2, ty = 2 Output: false

Example 3: Input: sx = 1, sy = 1, tx = 1, ty = 1 Output: true

Constraints:

1 <= sx, sy, tx, ty <= 109

==
Problem Number: 1788 URL: https://leetcode.com/problems/transform-to-chessboard Title: 782. Transform to Chessboard Problem Description: You are given an n x n binary grid board. In each move, you can swap any two rows with each other, or any two columns with each other. Return the minimum number of moves to transform the board into a chessboard board. If the task is impossible, return -1. A chessboard board is a board where no 0's and no 1's are 4-directionally adjacent.   Example 1:

Input: board = [[0,1,1,0],[0,1,1,0],[1,0,0,1],[1,0,0,1]] Output: 2 Explanation: One potential sequence of moves is shown. The first move swaps the first and second column. The second move swaps the second and third row.

Example 2:

Input: board = [[0,1],[1,0]] Output: 0 Explanation: Also note that the board with 0 in the top left corner, is also a valid chessboard.

Example 3:

Input: board = [[1,0],[1,0]] Output: -1 Explanation: No matter what sequence of moves you make, you cannot end with a valid chessboard.

Constraints:

n == board.length n == board[i].length 2 <= n <= 30 board[i][j] is either 0 or 1.

==
Problem Number: 1789 URL: https://leetcode.com/problems/preimage-size-of-factorial-zeroes-function Title: 793. Preimage Size of Factorial Zeroes Function Problem Description: Let f(x) be the number of zeroes at the end of x!. Recall that x! = 1 * 2 * 3 * ... * x and by convention, 0! = 1.

For example, f(3) = 0 because 3! = 6 has no zeroes at the end, while f(11) = 2 because 11! = 39916800 has two zeroes at the end.

Given an integer k, return the number of non-negative integers x have the property that f(x) = k.   Example 1: Input: k = 0 Output: 5 Explanation: 0!, 1!, 2!, 3!, and 4! end with k = 0 zeroes.

Example 2: Input: k = 5 Output: 0 Explanation: There is no x such that x! ends in k = 5 zeroes.

Example 3: Input: k = 3 Output: 5

Constraints:

0 <= k <= 109

==
Problem Number: 1790 URL: https://leetcode.com/problems/smallest-rotation-with-highest-score Title: 798. Smallest Rotation with Highest Score

Problem Description: You are given an array nums. You can rotate it by a non-negative integer k so that the array becomes [nums[k], nums[k + 1], ... nums[nums.length - 1], nums[0], nums[1], ..., nums[k-1]]. Afterward, any entries that are less than or equal to their index are worth one point.

For example, if we have nums = [2,4,1,3,0], and we rotate by k = 2, it becomes [1,3,0,2,4]. This is worth 3 points because 1 > 0 [no points], 3 > 1 [no points], 0 <= 2 [one point], 2 <= 3 [one point], 4 <= 4 [one point].

Return the rotation index k that corresponds to the highest score we can achieve if we rotated nums by it. If there are multiple answers, return the smallest such index k. Example 1: Input: nums = [2,3,1,4,0] Output: 3 Explanation: Scores for each k are listed below: k = 0, nums = [2,3,1,4,0], score 2 k = 1, nums = [3,1,4,0,2], score 3 k = 2, nums = [1,4,0,2,3], score 3 k = 3, nums = [4,0,2,3,1], score 4 k = 4, nums = [0,2,3,1,4], score 3 So we should choose k = 3, which has the highest score.

Example 2: Input: nums = [1,3,0,2,4] Output: 0 Explanation: nums will always have 3 points no matter how it shifts. So we will choose the smallest k, which is 0.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] < nums.length

==

Problem Number: 1791 URL: https://leetcode.com/problems/minimum-swaps-to-make-sequences-increasing Title: 801. Minimum Swaps To Make Sequences Increasing Problem Description: You are given two integer arrays of the same length nums1 and nums2. In one operation, you are allowed to swap nums1[i] with nums2[i].

For example, if nums1 = [1,2,3,8], and nums2 = [5,6,7,4], you can swap the element at i = 3 to obtain nums1 = [1,2,3,4] and nums2 = [5,6,7,8].

Return the minimum number of needed operations to make nums1 and nums2 strictly increasing. The test cases are generated so that the given input always makes it possible. An array arr is strictly increasing if and only if arr[0] < arr[1] < arr[2] < ... < arr[arr.length - 1]. Example 1: Input: nums1 = [1,3,5,4], nums2 = [1,2,3,7] Output: 1 Explanation: Swap nums1[3] and nums2[3]. Then the sequences are: nums1 = [1, 3, 5, 7] and nums2 = [1, 2, 3, 4] which are both strictly increasing.

Example 2: Input: nums1 = [0,3,5,8,9], nums2 = [2,1,4,6,9] Output: 1

Constraints:

2 <= nums1.length <= 105 nums2.length == nums1.length 0 <= nums1[i], nums2[i] <= 2 * 105

==

Problem Number: 1792 URL: https://leetcode.com/problems/bricks-falling-when-hit Title: 803. Bricks Falling When Hit Problem Description: You are given an m x n binary grid, where each 1 represents a brick and 0 represents an empty space. A brick is stable if:

It is directly connected to the top of the grid, or At least one other brick in its four adjacent cells is stable.

You are also given an array hits, which is a sequence of erasures we want to apply. Each time we want to erase the brick at the location hits[i] = (rowi, coli). The brick on that location (if it exists) will disappear. Some other bricks may no longer be stable because of that erasure and will fall. Once a brick falls, it is immediately erased from the grid (i.e., it does not land on other stable bricks). Return an array result, where each result[i] is the number of bricks that will fall after the ith erasure is applied. Note that an erasure may refer to a location with no brick, and if it does, no bricks drop. Example 1: Input: grid = [[1,0,0,0],[1,1,1,0]], hits = [[1,0]] Output: [2] Explanation: Starting with the grid: [[1,0,0,0], [1,1,1,0]] We erase the underlined brick at (1,0), resulting in the grid: [[1,0,0,0], [0,1,1,0]] The two underlined bricks are no longer stable as they are no longer connected to the top nor adjacent to another stable brick, so they will fall. The resulting grid is: [[1,0,0,0], [0,0,0,0]] Hence the result is [2].

Example 2: Input: grid = [[1,0,0,0],[1,1,0,0]], hits = [[1,1],[1,0]] Output: [0,0] Explanation: Starting with the grid: [[1,0,0,0], [1,1,0,0]] We erase the underlined brick at (1,1), resulting in the grid: [[1,0,0,0], [1,0,0,0]] All remaining bricks are still stable, so no bricks fall. The grid remains the same: [[1,0,0,0], [1,0,0,0]] Next, we erase the underlined brick at (1,0), resulting in the grid: [[1,0,0,0], [0,0,0,0]] Once again, all remaining bricks are still stable, so no bricks fall. Hence the result is [0,0].

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 200 grid[i][j] is 0 or 1. 1 <= hits.length <= 4 * 104 hits[i].length == 2 0 <= xi <= m - 1 0 <= yi <= n - 1 All (xi, yi) are unique.

==
Problem Number: 1793 URL: https://leetcode.com/problems/split-array-with-same-average Title: 805. Split Array With Same Average Problem Description: You are given an integer array nums. You should move each element of nums into one of the two arrays A and B such that A and B are non-empty, and average(A) == average(B). Return true if it is possible to achieve that and false otherwise. Note that for an array arr, average(arr) is the sum of all the elements of arr over the length of arr. Example 1: Input: nums = [1,2,3,4,5,6,7,8] Output: true Explanation: We can split the array into [1,4,5,8] and [2,3,6,7], and both of them have an average of 4.5.

Example 2: Input: nums = [3,1] Output: false

Constraints:

1 <= nums.length <= 30 0 <= nums[i] <= 104

==
Problem Number: 1794 URL: https://leetcode.com/problems/chalkboard-xor-game Title: 810. Chalkboard XOR Game Problem Description: You are given an array of integers nums represents the numbers written on a chalkboard. Alice and Bob take turns erasing exactly one number from the chalkboard, with Alice starting first. If erasing a number causes the bitwise XOR of all the elements of the chalkboard to become 0, then that player loses. The bitwise XOR of one element is that element itself, and the bitwise XOR of no elements is 0. Also, if any player starts their turn with the bitwise XOR of all the elements of the chalkboard equal to 0, then that player wins. Return true if and only if Alice wins the game, assuming both players play optimally. Example 1: Input: nums = [1,1,2] Output: false Explanation: Alice has two choices: erase 1 or erase 2. If she erases 1, the nums array becomes [1, 2]. The bitwise XOR of all the elements of the chalkboard is 1 XOR 2 = 3. Now Bob can remove any element he wants, because Alice will be the one to erase the last element and she will lose. If Alice erases 2 first, now nums become [1, 1]. The bitwise XOR of all the elements of the chalkboard is 1 XOR 1 = 0. Alice will lose.

Example 2: Input: nums = [0,1] Output: true

Example 3: Input: nums = [1,2,3] Output: true

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] < 216

==
Problem Number: 1795 URL: https://leetcode.com/problems/bus-routes Title: 815. Bus Routes Problem Description: You are given an array routes representing bus routes where routes[i] is a bus route that the ith bus repeats forever.

For example, if routes[0] = [1, 5, 7], this means that the 0th bus travels in the sequence 1 -> 5 -> 7 -> 1 -> 5 -> 7 -> 1 -> ... forever.

You will start at the bus stop source (You are not on any bus initially), and you want to go to the bus stop target. You can travel between bus stops by buses only. Return the least number of buses you must take to travel from source to target. Return -1 if it is not possible. Example 1: Input: routes = [[1,2,7],[3,6,7]], source = 1, target = 6 Output: 2 Explanation: The best strategy is take the first bus to the bus stop 7, then take the second bus to the bus stop 6.

Example 2: Input: routes = [[7,12],[4,5,15],[6],[15,19],[9,12,13]], source = 15, target = 12 Output: -1

Constraints:

1 <= routes.length <= 500. 1 <= routes[i].length <= 105 All the values of routes[i] are unique. sum(routes[i].length) <= 105 0 <= routes[i][j] < 106 0 <= source, target < 106

==
Problem Number: 1796 URL: https://leetcode.com/problems/race-car Title: 818. Race Car Problem Description: Your car starts at position 0 and speed +1 on an infinite number line. Your car can go into negative positions. Your car drives automatically according to a sequence of instructions 'A' (accelerate) and 'R' (reverse):

When you get an instruction 'A', your car does the following:

position += speed speed *= 2

When you get an instruction 'R', your car does the following:

If your speed is positive then speed = -1 otherwise speed = 1

Your position stays the same.

For example, after commands "AAR", your car goes to positions 0 --> 1 --> 3 --> 3, and your speed goes to 1 --> 2 --> 4 --> -1. Given a target position target, return the length of the shortest sequence of instructions to get there. Example 1: Input: target = 3 Output: 2 Explanation: The shortest instruction sequence is "AA". Your position goes from 0 --> 1 --> 3.

Example 2: Input: target = 6 Output: 5 Explanation: The shortest instruction sequence is "AAARA". Your position goes from 0 --> 1 --> 3 --> 7 --> 7 --> 6.

Constraints:

1 <= target <= 104

==
Problem Number: 1797 URL: https://leetcode.com/problems/making-a-large-island Title: 827. Making A Large Island Problem Description: You are given an n x n binary matrix grid. You are allowed to change at most one 0 to be 1. Return the size of the largest island in grid after applying this operation. An island is a 4-directionally connected group of 1s. Example 1: Input: grid = [[1,0],[0,1]] Output: 3 Explanation: Change one 0 to 1 and connect two 1s, then we get an island with area = 3.

Example 2: Input: grid = [[1,1],[1,0]] Output: 4 Explanation: Change the 0 to 1 and make the island bigger, only one island with area = 4. Example 3: Input: grid = [[1,1],[1,1]] Output: 4 Explanation: Can't change any 0 to 1, only one island with area = 4.

Constraints:

n == grid.length n == grid[i].length 1 <= n <= 500 grid[i][j] is either 0 or 1.

==

Problem Number: 1798 URL: https://leetcode.com/problems/count-unique-characters-of-all-substrings-of-a-given-string Title: 828. Count Unique Characters of All Substrings of a Given String Problem Description: Let's define a function countUniqueChars(s) that returns the number of unique characters in s.

For example, calling countUniqueChars(s) if s = "LEETCODE" then "L", "T", "C", "O", "D" are the unique characters since they appear only once in s, therefore countUniqueChars(s) = 5.

Given a string s, return the sum of countUniqueChars(t) where t is a substring of s. The test cases are generated such that the answer fits in a 32-bit integer. Notice that some substrings can be repeated so in this case you have to count the repeated ones too. Example 1: Input: s = "ABC" Output: 10 Explanation: All possible substrings are: "A","B","C","AB","BC" and "ABC". Every substring is composed with only unique letters. Sum of lengths of all substring is 1 + 1 + 1 + 2 + 2 + 3 = 10

Example 2: Input: s = "ABA" Output: 8 Explanation: The same as example 1, except countUniqueChars("ABA") = 1.

Example 3: Input: s = "LEETCODE" Output: 92

Constraints:

1 <= s.length <= 105 s consists of uppercase English letters only.

==

Problem Number: 1799 URL: https://leetcode.com/problems/consecutive-numbers-sum Title: 829. Consecutive Numbers Sum Problem Description: Given an integer n, return the number of ways you can write n as the sum of consecutive positive integers. Example 1: Input: n = 5 Output: 2 Explanation: 5 = 2 + 3

Example 2: Input: n = 9 Output: 3 Explanation: 9 = 4 + 5 = 2 + 3 + 4

Example 3: Input: n = 15 Output: 4 Explanation: 15 = 8 + 7 = 4 + 5 + 6 = 1 + 2 + 3 + 4 + 5

Constraints:

1 <= n <= 109

==

Problem Number: 1800 URL: https://leetcode.com/problems/sum-of-distances-in-tree Title: 834. Sum of Distances in Tree Problem Description: There is an undirected connected tree with n nodes labeled from 0 to n - 1 and n - 1 edges. You are given the integer n and the array edges where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. Return an array answer of length n where answer[i] is the sum of the distances between the ith node in the tree and all other nodes. Example 1:

Input: n = 6, edges = [[0,1],[0,2],[2,3],[2,4],[2,5]] Output: [8,12,6,10,10,10] Explanation: The tree is shown above. We can see that dist(0,1) + dist(0,2) + dist(0,3) + dist(0,4) + dist(0,5) equals 1 + 1 + 2 + 2 + 2 = 8. Hence, answer[0] = 8, and so on.

Example 2:

Input: n = 1, edges = [] Output: [0]

Example 3:

Input: n = 2, edges = [[1,0]] Output: [1,1]

Constraints:

1 <= n <= 3 * 104 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi The given input represents a valid tree.

== Problem Number: 1801 URL: https://leetcode.com/problems/similar-string-groups Title: 839. Similar String Groups Problem Description: Two strings, X and Y, are considered similar if either they are identical or we can make them equivalent by swapping at most two letters (in distinct positions) within the string X. For example, "tars" and "rats" are similar (swapping at positions 0 and 2), and "rats" and "arts" are similar, but "star" is not similar to "tars", "rats", or "arts". Together, these form two connected groups by similarity: {"tars", "rats", "arts"} and {"star"}. Notice that "tars" and "arts" are in the same group even though they are not similar. Formally, each group is such that a word is in the group if and only if it is similar to at least one other word in the group. We are given a list strs of strings where every string in strs is an anagram of every other string in strs. How many groups are there? Example 1: Input: strs = ["tars","rats","arts","star"] Output: 2

Example 2: Input: strs = ["omv","ovm"] Output: 1

Constraints:

1 <= strs.length <= 300 1 <= strs[i].length <= 300 strs[i] consists of lowercase letters only. All words in strs have the same length and are anagrams of each other.

== Problem Number: 1802 URL: https://leetcode.com/problems/guess-the-word Title: 843. Guess the Word Problem Description: You are given an array of unique strings words where words[i] is six letters long. One word of words was chosen as a secret word. You are also given the helper object Master. You may call Master.guess(word) where word is a six-letter-long string, and it must be from words. Master.guess(word) returns:

-1 if word is not from words, or an integer representing the number of exact matches (value and position) of your guess to the secret word.

There is a parameter allowedGuesses for each test case where allowedGuesses is the maximum number of times you can call Master.guess(word). For each test case, you should call Master.guess with the secret word without exceeding the maximum number of allowed guesses. You will get:

"Either you took too many guesses, or you did not find the secret word." if you called Master.guess more than allowedGuesses times or if you did not call Master.guess with the secret word, or "You guessed the secret word correctly." if you called Master.guess with the secret word with the number of calls to Master.guess less than or equal to allowedGuesses.

The test cases are generated such that you can guess the secret word with a reasonable strategy (other than using the bruteforce method). Example 1: Input: secret = "acckzz", words = ["acckzz","ccbazz","eiowzz","abcczz"], allowedGuesses = 10 Output: You guessed the secret word correctly. Explanation: master.guess("aaaaaa") returns -1, because "aaaaaa" is not in wordlist. master.guess("acckzz") returns 6, because "acckzz" is secret and has all 6 matches. master.guess("ccbazz") returns 3, because "ccbazz" has 3 matches. master.guess("eiowzz") returns 2, because "eiowzz" has 2 matches. master.guess("abcczz") returns 4, because "abcczz" has 4 matches. We made 5 calls to master.guess, and one of them was the secret, so we pass the test case.

Example 2: Input: secret = "hamada", words = ["hamada","khaled"], allowedGuesses = 10 Output: You guessed the secret word correctly. Explanation: Since there are two words, you can guess both.

Constraints:

1 <= words.length <= 100 words[i].length == 6 words[i] consist of lowercase English letters. All the strings of wordlist are unique. secret exists in words. 10 <= allowedGuesses <= 30

==
Problem Number: 1803 URL: https://leetcode.com/problems/shortest-path-visiting-all-nodes Title: 847. Shortest Path Visiting All Nodes Problem Description: You have an undirected, connected graph of n nodes labeled from 0 to n - 1. You are given an array graph where graph[i] is a list of all the nodes connected with node i by an edge. Return the length of the shortest path that visits every node. You may start and stop at any node, you may revisit nodes multiple times, and you may reuse edges. Example 1:

Input: graph = [[1,2,3],[0],[0],[0]] Output: 4 Explanation: One possible path is [1,0,2,0,3]

Example 2:

Input: graph = [[1],[0,2,4],[1,3,4],[2],[1,2]] Output: 4 Explanation: One possible path is [0,1,4,2,3]

Constraints:

n == graph.length 1 <= n <= 12 0 <= graph[i].length < n graph[i] does not contain i. If graph[a] contains b, then graph[b] contains a. The input graph is always connected.

==

Problem Number: 1804 URL: https://leetcode.com/problems/rectangle-area-ii Title: 850. Rectangle Area II Problem Description: You are given a 2D array of axis-aligned rectangles. Each rectangle[i] = [xi1, yi1, xi2, yi2] denotes the ith rectangle where (xi1, yi1) are the coordinates of the bottom-left corner, and (xi2, yi2) are the coordinates of the top-right corner. Calculate the total area covered by all rectangles in the plane. Any area covered by two or more rectangles should only be counted once. Return the total area. Since the answer may be too large, return it modulo 109 + 7. Example 1:

Input: rectangles = [[0,0,2,2],[1,0,2,3],[1,0,3,1]] Output: 6 Explanation: A total area of 6 is covered by all three rectangles, as illustrated in the picture. From (1,1) to (2,2), the green and red rectangles overlap. From (1,0) to (2,3), all three rectangles overlap.

Example 2: Input: rectangles = [[0,0,1000000000,1000000000]] Output: 49 Explanation: The answer is 1018 modulo (109 + 7), which is 49.

Constraints:

1 <= rectangles.length <= 200 rectanges[i].length == 4 0 <= xi1, yi1, xi2, yi2 <= 109 xi1 <= xi2 yi1 <= yi2

==

Problem Number: 1805 URL: https://leetcode.com/problems/k-similar-strings Title: 854. K-Similar Strings Problem Description: Strings s1 and s2 are k-similar (for some non-negative integer k) if we can swap the positions of two letters in s1 exactly k times so that the resulting string equals s2. Given two anagrams s1 and s2, return the smallest k for which s1 and s2 are k-similar. Example 1: Input: s1 = "ab", s2 = "ba" Output: 1 Explanation: The two string are 1-similar because we can use one swap to change s1 to s2: "ab" --> "ba".

Example 2: Input: s1 = "abc", s2 = "bca" Output: 2 Explanation: The two strings are 2-similar because we can use two swaps to change s1 to s2: "abc" --> "bac" --> "bca".

Constraints:

1 <= s1.length <= 20 s2.length == s1.length s1 and s2 contain only lowercase letters from the set {'a', 'b', 'c', 'd', 'e', 'f'}. s2 is an anagram of s1.

==

Problem Number: 1806 URL: https://leetcode.com/problems/minimum-cost-to-hire-k-workers Title: 857. Minimum Cost to Hire K Workers Problem Description: There are n workers. You are given two integer arrays quality and wage where quality[i] is the quality of the ith worker and wage[i] is the minimum wage expectation for the ith worker. We want to hire exactly k workers to form a paid group. To hire a group of k workers, we must pay them according to the following rules:

Every worker in the paid group should be paid in the ratio of their quality compared to other workers in the paid group. Every worker in the paid group must be paid at least their minimum wage expectation.

Given the integer k, return the least amount of money needed to form a paid group satisfying the above conditions. Answers within 10-5 of the actual answer will be accepted. Example 1: Input: quality = [10,20,5], wage = [70,50,30], k = 2 Output: 105.00000 Explanation: We pay 70 to 0th worker and 35 to 2nd worker.

Example 2: Input: quality = [3,1,10,10,1], wage = [4,8,2,2,7], k = 3 Output: 30.66667 Explanation: We pay 4 to 0th worker, 13.33333 to 2nd and 3rd workers separately.

Constraints:

n == quality.length == wage.length 1 <= k <= n <= 104 1 <= quality[i], wage[i] <= 104

==
Problem Number: 1807 URL: https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k Title: 862. Shortest Subarray with Sum at Least K Problem Description: Given an integer array nums and an integer k, return the length of the shortest non-empty subarray of nums with a sum of at least k. If there is no such subarray, return -1. A subarray is a contiguous part of an array. Example 1: Input: nums = [1], k = 1 Output: 1 Example 2: Input: nums = [1,2], k = 4 Output: -1 Example 3: Input: nums = [2,-1,2], k = 3 Output: 3

Constraints:

1 <= nums.length <= 105 -105 <= nums[i] <= 105 1 <= k <= 109

==
Problem Number: 1808 URL: https://leetcode.com/problems/shortest-path-to-get-all-keys Title: 864. Shortest Path to Get All Keys Problem Description: You are given an m x n grid grid where:

'.' is an empty cell. '#' is a wall. '@' is the starting point. Lowercase letters represent keys. Uppercase letters represent locks.

You start at the starting point and one move consists of walking one space in one of the four cardinal directions. You cannot walk outside the grid, or walk into a wall. If you walk over a key, you can pick it up and you cannot walk over a lock unless you have its corresponding key. For some 1 <= k <= 6, there is exactly one lowercase and one uppercase letter of the first k letters of the English alphabet in the grid. This means that there is exactly one key for each lock, and one lock for each key; and also that the letters used to represent the keys and locks were chosen in the same order as the English alphabet. Return the lowest number of moves to acquire all keys. If it is impossible, return -1. Example 1:

Input: grid = ["@.a..","###.#","b.A.B"] Output: 8 Explanation: Note that the goal is to obtain all the keys not to open all the locks.

Example 2:

Input: grid = ["@..aA","..B#.","....b"] Output: 6

Example 3:

Input: grid = ["@Aa"] Output: -1

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 30 grid[i][j] is either an English letter, '.', '#', or '@'. There is exactly one '@' in the grid. The number of keys in the grid is in the range [1, 6]. Each key in the grid is unique. Each key in the grid has a matching lock.

==
Problem Number: 1809 URL: https://leetcode.com/problems/minimum-number-of-refueling-stops Title: 871. Minimum Number of Refueling Stops Problem Description: A car travels from a starting position to a destination which is target miles east of the starting position. There are gas stations along the way. The gas stations are represented as an array stations where stations[i] = [positioni, fueli] indicates that the ith gas station is positioni miles east of the starting position and has fueli liters of gas. The car starts with an infinite tank of gas, which initially has startFuel liters of fuel in it. It uses one liter of gas per one mile that it drives. When the car reaches a gas station, it may stop and refuel, transferring all the gas from the station into the car. Return the minimum number of refueling stops the car must make in order to reach its destination. If it cannot reach the destination, return -1. Note that if the car reaches a gas station with 0 fuel left, the car can still refuel there. If the car reaches the destination with 0 fuel left, it is still considered to have arrived. Example 1: Input: target = 1, startFuel = 1, stations = [] Output: 0 Explanation: We can reach the target without refueling.

Example 2: Input: target = 100, startFuel = 1, stations = [[10,100]] Output: -1 Explanation: We can not reach the target (or even the first gas station).

Example 3: Input: target = 100, startFuel = 10, stations = [[10,60],[20,30],[30,30],[60,40]] Output: 2 Explanation: We start with 10 liters of fuel. We drive to position 10, expending 10 liters of fuel. We refuel from 0 liters to 60 liters of gas. Then, we drive from position 10 to position 60 (expending 50 liters of fuel), and refuel from 10 liters to 50 liters of gas. We then drive to and reach the target. We made 2 refueling stops along the way, so we return 2.

Constraints:

1 <= target, startFuel <= 109 0 <= stations.length <= 500 1 <= positioni < positioni+1 < target 1 <= fueli < 109

==
Problem Number: 1810 URL: https://leetcode.com/problems/nth-magical-number Title: 878. Nth Magical Number Problem Description: A positive integer is magical if it is divisible by either a or b. Given the three integers n, a, and b, return the nth magical number. Since the answer may be very large, return it modulo 109 + 7.   Example 1: Input: n = 1, a = 2, b = 3 Output: 2

Example 2: Input: n = 4, a = 2, b = 3 Output: 6

Constraints:

1 <= n <= 109 2 <= a, b <= 4 * 104

==
Problem Number: 1811 URL: https://leetcode.com/problems/profitable-schemes Title: 879. Profitable Schemes Problem Description: There is a group of n members, and a list of various crimes they could commit. The ith crime generates a profit[i] and requires group[i] members to participate in it. If a member participates in one crime, that member can't participate in another crime. Let's call a profitable scheme any subset of these crimes that generates at least minProfit profit, and the total number of members participating in that subset of crimes is at most n. Return the number of schemes that can be chosen. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: n = 5, minProfit = 3, group = [2,2], profit = [2,3] Output: 2 Explanation: To make a profit of at least 3, the group could either commit crimes 0 and 1, or just crime 1. In total, there are 2 schemes. Example 2: Input: n = 10, minProfit = 5, group = [2,3,5], profit = [6,7,8] Output: 7 Explanation: To make a profit of at least 5, the group could commit any crimes, as long as they commit one. There are 7 possible schemes: (0), (1), (2), (0,1), (0,2), (1,2), and (0,1,2).   Constraints:

1 <= n <= 100 0 <= minProfit <= 100 1 <= group.length <= 100 1 <= group[i] <= 100 profit.length == group.length 0 <= profit[i] <= 100

==
Problem Number: 1812 URL: https://leetcode.com/problems/reachable-nodes-in-subdivided-graph Title: 882. Reachable Nodes In Subdivided Graph Problem Description: You are given an undirected graph (the "original graph") with n nodes labeled from 0 to n - 1. You decide to subdivide each edge in the graph into a chain of nodes, with the number of new nodes varying between each edge. The graph is given as a 2D array of edges where edges[i] = [ui, vi, cnti] indicates that there is an edge between nodes ui and vi in the original graph, and cnti is the total number of new nodes that you will subdivide the edge into. Note that cnti == 0 means you will not subdivide the edge. To subdivide the edge [ui, vi], replace it with (cnti + 1) new edges and cnti new

nodes. The new nodes are x1, x2, ..., xcnti, and the new edges are [ui, x1], [x1, x2], [x2, x3], ..., [xcnti-1, xcnti], [xcnti, vi]. In this new graph, you want to know how many nodes are reachable from the node 0, where a node is reachable if the distance is maxMoves or less. Given the original graph and maxMoves, return the number of nodes that are reachable from node 0 in the new graph. Example 1:

Input: edges = [[0,1,10],[0,2,1],[1,2,2]], maxMoves = 6, n = 3 Output: 13 Explanation: The edge subdivisions are shown in the image above. The nodes that are reachable are highlighted in yellow.

Example 2: Input: edges = [[0,1,4],[1,2,6],[0,2,8],[1,3,1]], maxMoves = 10, n = 4 Output: 23

Example 3: Input: edges = [[1,2,4],[1,4,5],[1,3,1],[2,3,4],[3,4,5]], maxMoves = 17, n = 5 Output: 1 Explanation: Node 0 is disconnected from the rest of the graph, so only node 0 is reachable.

Constraints:

0 <= edges.length <= min(n * (n - 1) / 2, 104) edges[i].length == 3 0 <= ui < vi < n There are no multiple edges in the graph. 0 <= cnti <= 104 0 <= maxMoves <= 109 1 <= n <= 3000

==
Problem Number: 1813 URL: https://leetcode.com/problems/super-egg-drop Title: 887. Super Egg Drop Problem Description: You are given k identical eggs and you have access to a building with n floors labeled from 1 to n. You know that there exists a floor f where 0 <= f <= n such that any egg dropped at a floor higher than f will break, and any egg dropped at or below floor f will not break. Each move, you may take an unbroken egg and drop it from any floor x (where 1 <= x <= n). If the egg breaks, you can no longer use it. However, if the egg does not break, you may reuse it in future moves. Return the minimum number of moves that you need to determine with certainty what the value of f is. Example 1: Input: k = 1, n = 2 Output: 2 Explanation: Drop the egg from floor 1. If it breaks, we know that f = 0. Otherwise, drop the egg from floor 2. If it breaks, we know that f = 1. If it does not break, then we know f = 2. Hence, we need at minimum 2 moves to determine with certainty what the value of f is.

Example 2: Input: k = 2, n = 6 Output: 3

Example 3: Input: k = 3, n = 14 Output: 4

Constraints:

1 <= k <= 100 1 <= n <= 104

==
Problem Number: 1814 URL: https://leetcode.com/problems/sum-of-subsequence-widths Title: 891. Sum of Subsequence Widths Problem

Description: The width of a sequence is the difference between the maximum and minimum elements in the sequence. Given an array of integers nums, return the sum of the widths of all the non-empty subsequences of nums. Since the answer may be very large, return it modulo 109 + 7. A subsequence is a sequence that can be derived from an array by deleting some or no elements without changing the order of the remaining elements. For example, [3,6,2,7] is a subsequence of the array [0,3,1,6,2,2,7]. Example 1: Input: nums = [2,1,3] Output: 6 Explanation: The subsequences are [1], [2], [3], [2,1], [2,3], [1,3], [2,1,3]. The corresponding widths are 0, 0, 0, 1, 1, 2, 2. The sum of these widths is 6.

Example 2: Input: nums = [2] Output: 0

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105

==

Problem Number: 1815 URL: https://leetcode.com/problems/maximum-frequency-stack Title: 895. Maximum Frequency Stack Problem Description: Design a stack-like data structure to push elements to the stack and pop the most frequent element from the stack. Implement the FreqStack class:

FreqStack() constructs an empty frequency stack. void push(int val) pushes an integer val onto the top of the stack. int pop() removes and returns the most frequent element in the stack.

If there is a tie for the most frequent element, the element closest to the stack's top is removed and returned.

Example 1: Input ["FreqStack", "push", "push", "push", "push", "push", "push", "pop", "pop", "pop", "pop"] [[], [5], [7], [5], [7], [4], [5], [], [], [], []] Output [null, null, null, null, null, null, null, 5, 7, 5, 4]

Explanation FreqStack freqStack = new FreqStack(); freqStack.push(5); // The stack is [5] freqStack.push(7); // The stack is [5,7] freqStack.push(5); // The stack is [5,7,5] freqStack.push(7); // The stack is [5,7,5,7] freqStack.push(4); // The stack is [5,7,5,7,4] freqStack.push(5); // The stack is [5,7,5,7,4,5] freqStack.pop(); // return 5, as 5 is the most frequent. The stack becomes [5,7,5,7,4]. freqStack.pop(); // return 7, as 5 and 7 is the most frequent, but 7 is closest to the top. The stack becomes [5,7,5,4]. freqStack.pop(); // return 5, as 5 is the most frequent. The stack becomes [5,7,4]. freqStack.pop(); // return 4, as 4, 5 and 7 is the most frequent, but 4 is closest to the top. The stack becomes [5,7].

Constraints:

0 <= val <= 109 At most 2 * 104 calls will be made to push and pop. It is guaranteed that there will be at least one element in the stack before calling pop.

==

Problem Number: 1816 URL: https://leetcode.com/problems/orderly-queue Title: 899. Orderly Queue Problem Description: You are given a string s and an integer k. You can choose one of the first k letters of s and append it at the end of the string. Return the lexicographically smallest string you could have after applying the mentioned step any number of moves. Example 1: Input: s = "cba", k = 1 Output: "acb" Explanation: In the first move, we move the 1st character 'c' to the end, obtaining the string "bac". In the second move, we move the 1st character 'b' to the end, obtaining the final result "acb".

Example 2: Input: s = "baaca", k = 3 Output: "aaabc" Explanation: In the first move, we move the 1st character 'b' to the end, obtaining the string "aacab". In the second move, we move the 3rd character 'c' to the end, obtaining the final result "aaabc".

Constraints:

1 <= k <= s.length <= 1000 s consist of lowercase English letters.

==

Problem Number: 1817 URL: https://leetcode.com/problems/numbers-at-most-n-given-digit-set Title: 902. Numbers At Most N Given Digit Set Problem Description: Given an array of digits which is sorted in non-decreasing order. You can write numbers using each digits[i] as many times as we want. For example, if digits = ['1','3','5'], we may write numbers such as '13', '551', and '1351315'. Return the number of positive integers that can be generated that are less than or equal to a given integer n. Example 1: Input: digits = ["1","3","5","7"], n = 100 Output: 20 Explanation: The 20 numbers that can be written are: 1, 3, 5, 7, 11, 13, 15, 17, 31, 33, 35, 37, 51, 53, 55, 57, 71, 73, 75, 77.

Example 2: Input: digits = ["1","4","9"], n = 1000000000 Output: 29523 Explanation: We can write 3 one digit numbers, 9 two digit numbers, 27 three digit numbers, 81 four digit numbers, 243 five digit numbers, 729 six digit numbers, 2187 seven digit numbers, 6561 eight digit numbers, and 19683 nine digit numbers. In total, this is 29523 integers that can be written using the digits array.

Example 3: Input: digits = ["7"], n = 8 Output: 1

Constraints:

1 <= digits.length <= 9 digits[i].length == 1 digits[i] is a digit from '1' to '9'. All the values in digits are unique. digits is sorted in non-decreasing order. 1 <= n <= 109

==

Problem Number: 1818 URL: https://leetcode.com/problems/valid-permutations-for-di-sequence Title: 903. Valid Permutations for DI Sequence Problem Description: You are given a string s of length n where s[i] is either:

'D' means decreasing, or 'I' means increasing.

A permutation perm of n + 1 integers of all the integers in the range [0, n] is called a valid permutation if for all valid i:

If s[i] == 'D', then perm[i] > perm[i + 1], and If s[i] == 'I', then perm[i] < perm[i + 1].

Return the number of valid permutations perm. Since the answer may be large, return it modulo 109 + 7. Example 1: Input: s = "DID" Output: 5 Explanation: The 5 valid permutations of (0, 1, 2, 3) are: (1, 0, 3, 2) (2, 0, 3, 1) (2, 1, 3, 0) (3, 0, 2, 1) (3, 1, 2, 0)

Example 2: Input: s = "D" Output: 1

Constraints:

n == s.length 1 <= n <= 200 s[i] is either 'I' or 'D'.

==
Problem Number: 1819 URL: https://leetcode.com/problems/super-palindromes Title: 906. Super Palindromes Problem Description: Let's say a positive integer is a super-palindrome if it is a palindrome, and it is also the square of a palindrome. Given two positive integers left and right represented as strings, return the number of super-palindromes integers in the inclusive range [left, right]. Example 1: Input: left = "4", right = "1000" Output: 4 Explanation: 4, 9, 121, and 484 are superpalindromes. Note that 676 is not a superpalindrome: 26 * 26 = 676, but 26 is not a palindrome.

Example 2: Input: left = "1", right = "2" Output: 1

Constraints:

1 <= left.length, right.length <= 18 left and right consist of only digits. left and right cannot have leading zeros. left and right represent integers in the range [1, 1018 - 1]. left is less than or equal to right.

==
Problem Number: 1820 URL: https://leetcode.com/problems/cat-and-mouse Title: 913. Cat and Mouse Problem Description: A game on an undirected graph is played by two players, Mouse and Cat, who alternate turns. The graph is given as follows: graph[a] is a list of all nodes b such that ab is an edge of the graph. The mouse starts at node 1 and goes first, the cat starts at node 2 and goes second, and there is a hole at node 0. During each player's turn, they must travel along one edge of the graph that meets where they are. For example, if the Mouse is at node 1, it must travel to any node in graph[1]. Additionally, it is not allowed for the Cat to travel to the Hole (node 0.) Then, the game can end in three ways:

If ever the Cat occupies the same node as the Mouse, the Cat wins. If ever the Mouse reaches the Hole, the Mouse wins. If ever a position is repeated (i.e., the players are in the same position as a previous turn, and it is the same player's turn to move), the game is a draw.

Given a graph, and assuming both players play optimally, return

1 if the mouse wins the game, 2 if the cat wins the game, or 0 if the game is a draw.

Example 1:

Input: graph = [[2,5],[3],[0,4,5],[1,4,5],[2,3],[0,2,3]] Output: 0

Example 2:

Input: graph = [[1,3],[0],[3],[0,2]] Output: 1

Constraints:

3 <= graph.length <= 50 1 <= graph[i].length < graph.length 0 <= graph[i][j] < graph.length graph[i][j] != i graph[i] is unique. The mouse and the cat can always move.

==
Problem Number: 1821 URL: https://leetcode.com/problems/number-of-music-playlists Title: 920. Number of Music Playlists Problem Description: Your music player contains n different songs. You want to listen to goal songs (not necessarily different) during your trip. To avoid boredom, you will create a playlist so that:

Every song is played at least once. A song can only be played again only if k other songs have been played.

Given n, goal, and k, return the number of possible playlists that you can create. Since the answer can be very large, return it modulo 109 + 7. Example 1: Input: n = 3, goal = 3, k = 1 Output: 6 Explanation: There are 6 possible playlists: [1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], and [3, 2, 1].

Example 2: Input: n = 2, goal = 3, k = 0 Output: 6 Explanation: There are 6 possible playlists: [1, 1, 2], [1, 2, 1], [2, 1, 1], [2, 2, 1], [2, 1, 2], and [1, 2, 2].

Example 3: Input: n = 2, goal = 3, k = 1 Output: 2 Explanation: There are 2 possible playlists: [1, 2, 1] and [2, 1, 2].

Constraints:

0 <= k < n <= goal <= 100

==
Problem Number: 1822 URL: https://leetcode.com/problems/minimize-malware-spread Title: 924. Minimize Malware Spread Problem Description: You are given a network of n nodes represented as an n x n adjacency matrix graph, where the ith node is directly connected to the jth node if graph[i][j] == 1. Some nodes initial are initially infected by malware. Whenever two nodes are directly connected, and at least one of those two nodes is infected by malware, both nodes will be infected by malware. This spread of malware will continue until no more nodes can be infected in this manner. Suppose

M(initial) is the final number of nodes infected with malware in the entire network after the spread of malware stops. We will remove exactly one node from initial. Return the node that, if removed, would minimize M(initial). If multiple nodes could be removed to minimize M(initial), return such a node with the smallest index. Note that if a node was removed from the initial list of infected nodes, it might still be infected later due to the malware spread. Example 1: Input: graph = [[1,1,0],[1,1,0],[0,0,1]], initial = [0,1] Output: 0 Example 2: Input: graph = [[1,0,0],[0,1,0],[0,0,1]], initial = [0,2] Output: 0 Example 3: Input: graph = [[1,1,1],[1,1,1],[1,1,1]], initial = [1,2] Output: 1

Constraints:

n == graph.length n == graph[i].length 2 <= n <= 300 graph[i][j] is 0 or 1. graph[i][j] == graph[j][i] graph[i][i] == 1 1 <= initial.length <= n 0 <= initial[i] <= n - 1 All the integers in initial are unique.

==
Problem Number: 1823 URL: https://leetcode.com/problems/three-equal-parts Title: 927. Three Equal Parts Problem Description: You are given an array arr which consists of only zeros and ones, divide the array into three non-empty parts such that all of these parts represent the same binary value. If it is possible, return any [i, j] with i + 1 < j, such that:

arr[0], arr[1], ..., arr[i] is the first part, arr[i + 1], arr[i + 2], ..., arr[j - 1] is the second part, and arr[j], arr[j + 1], ..., arr[arr.length - 1] is the third part. All three parts have equal binary values.

If it is not possible, return [-1, -1]. Note that the entire part is used when considering what binary value it represents. For example, [1,1,0] represents 6 in decimal, not 3. Also, leading zeros are allowed, so [0,1,1] and [1,1] represent the same value. Example 1: Input: arr = [1,0,1,0,1] Output: [0,3] Example 2: Input: arr = [1,1,0,1,1] Output: [-1,-1] Example 3: Input: arr = [1,1,0,0,1] Output: [0,2]

Constraints:

3 <= arr.length <= 3 * 104 arr[i] is 0 or 1

==
Problem Number: 1824 URL: https://leetcode.com/problems/minimize-malware-spread-ii Title: 928. Minimize Malware Spread II Problem Description: You are given a network of n nodes represented as an n x n adjacency matrix graph, where the ith node is directly connected to the jth node if graph[i][j] == 1. Some nodes initial are initially infected by malware. Whenever two nodes are directly connected, and at least one of those two nodes is infected by malware, both nodes will be infected by malware. This spread of malware will continue until no more nodes can be infected in this manner. Suppose M(initial) is the final number of nodes infected with malware in the entire network after the spread of malware stops. We will remove exactly one node from initial, completely removing it and any connections from this node to any other node.

Return the node that, if removed, would minimize M(initial). If multiple nodes could be removed to minimize M(initial), return such a node with the smallest index. Example 1: Input: graph = [[1,1,0],[1,1,0],[0,0,1]], initial = [0,1] Output: 0 Example 2: Input: graph = [[1,1,0],[1,1,1],[0,1,1]], initial = [0,1] Output: 1 Example 3: Input: graph = [[1,1,0,0],[1,1,1,0],[0,1,1,1],[0,0,1,1]], initial = [0,1] Output: 1

Constraints:

n == graph.length n == graph[i].length 2 <= n <= 300 graph[i][j] is 0 or 1. graph[i][j] == graph[j][i] graph[i][i] == 1 1 <= initial.length < n 0 <= initial[i] <= n - 1 All the integers in initial are unique.

==

Problem Number: 1825 URL: https://leetcode.com/problems/stamping-the-sequence Title: 936. Stamping The Sequence Problem Description: You are given two strings stamp and target. Initially, there is a string s of length target.length with all s[i] == '?'. In one turn, you can place stamp over s and replace every letter in the s with the corresponding letter from stamp.

For example, if stamp = "abc" and target = "abcba", then s is "?????" initially. In one turn you can:

place stamp at index 0 of s to obtain "abc??", place stamp at index 1 of s to obtain "?abc?", or place stamp at index 2 of s to obtain "??abc".

Note that stamp must be fully contained in the boundaries of s in order to stamp (i.e., you cannot place stamp at index 3 of s).

We want to convert s to target using at most 10 * target.length turns. Return an array of the index of the left-most letter being stamped at each turn. If we cannot obtain target from s within 10 * target.length turns, return an empty array. Example 1: Input: stamp = "abc", target = "ababc" Output: [0,2] Explanation: Initially s = "?????". - Place stamp at index 0 to get "abc??". - Place stamp at index 2 to get "ababc". [1,0,2] would also be accepted as an answer, as well as some other answers.

Example 2: Input: stamp = "abca", target = "aabcaca" Output: [3,0,1] Explanation: Initially s = "???????". - Place stamp at index 3 to get "???abca". - Place stamp at index 0 to get "abcabca". - Place stamp at index 1 to get "aabcaca".

Constraints:

1 <= stamp.length <= target.length <= 1000 stamp and target consist of lowercase English letters.

==

Problem Number: 1826 URL: https://leetcode.com/problems/distinct-subsequences-ii Title: 940. Distinct Subsequences II Problem Description: Given a string s, return the number of distinct non-empty subsequences of s.

Since the answer may be very large, return it modulo 109 + 7. A subsequence of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., "ace" is a subsequence of "abcde" while "aec" is not. Example 1: Input: s = "abc" Output: 7 Explanation: The 7 distinct subsequences are "a", "b", "c", "ab", "ac", "bc", and "abc".

Example 2: Input: s = "aba" Output: 6 Explanation: The 6 distinct subsequences are "a", "b", "ab", "aa", "ba", and "aba".

Example 3: Input: s = "aaa" Output: 3 Explanation: The 3 distinct subsequences are "a", "aa" and "aaa".

Constraints:

1 <= s.length <= 2000 s consists of lowercase English letters.

==
Problem Number: 1827 URL: https://leetcode.com/problems/find-the-shortest-superstring Title: 943. Find the Shortest Superstring Problem Description: Given an array of strings words, return the smallest string that contains each string in words as a substring. If there are multiple valid strings of the smallest length, return any of them. You may assume that no string in words is a substring of another string in words. Example 1: Input: words = ["alex","loves","leetcode"] Output: "alexlovesleetcode" Explanation: All permutations of "alex","loves","leetcode" would also be accepted.

Example 2: Input: words = ["catg","ctaagt","gcta","ttca","atgcatc"] Output: "gctaagttcatgcatc"

Constraints:

1 <= words.length <= 12 1 <= words[i].length <= 20 words[i] consists of lowercase English letters. All the strings of words are unique.

==
Problem Number: 1828 URL: https://leetcode.com/problems/largest-component-size-by-common-factor Title: 952. Largest Component Size by Common Factor Problem Description: You are given an integer array of unique positive integers nums. Consider the following graph:

There are nums.length nodes, labeled nums[0] to nums[nums.length - 1], There is an undirected edge between nums[i] and nums[j] if nums[i] and nums[j] share a common factor greater than 1.

Return the size of the largest connected component in the graph. Example 1:

Input: nums = [4,6,15,35] Output: 4

Example 2:

Input: nums = [20,50,9,63] Output: 2

Example 3:

Input: nums = [2,3,6,7,4,12,21,39] Output: 8

Constraints:

1 <= nums.length <= 2 * 104 1 <= nums[i] <= 105 All the values of nums are unique.

==

Problem Number: 1829 URL: https://leetcode.com/problems/tallest-billboard Title: 956. Tallest Billboard Problem Description: You are installing a billboard and want it to have the largest height. The billboard will have two steel supports, one on each side. Each steel support must be an equal height. You are given a collection of rods that can be welded together. For example, if you have rods of lengths 1, 2, and 3, you can weld them together to make a support of length 6. Return the largest possible height of your billboard installation. If you cannot support the billboard, return 0. Example 1: Input: rods = [1,2,3,6] Output: 6 Explanation: We have two disjoint subsets {1,2,3} and {6}, which have the same sum = 6.

Example 2: Input: rods = [1,2,3,4,5,6] Output: 10 Explanation: We have two disjoint subsets {2,3,5} and {4,6}, which have the same sum = 10.

Example 3: Input: rods = [1,2] Output: 0 Explanation: The billboard cannot be supported, so we return 0.

Constraints:

1 <= rods.length <= 20 1 <= rods[i] <= 1000 sum(rods[i]) <= 5000

==

Problem Number: 1830 URL: https://leetcode.com/problems/delete-columns-to-make-sorted-iii Title: 960. Delete Columns to Make Sorted III Problem Description: You are given an array of n strings strs, all of the same length. We may choose any deletion indices, and we delete all the characters in those indices for each string. For example, if we have strs = ["abcdef","uvwxyz"] and deletion indices {0, 2, 3}, then the final array after deletions is ["bef", "vyz"]. Suppose we chose a set of deletion indices answer such that after deletions, the final array has every string (row) in lexicographic order. (i.e., (strs[0][0] <= strs[0][1] <= ... <= strs[0][strs[0].length - 1]), and (strs[1][0] <= strs[1][1] <= ... <= strs[1][strs[1].length - 1]), and so on). Return the minimum possible value of answer.length. Example 1: Input: strs = ["babca","bbazb"] Output: 3 Explanation: After deleting columns 0, 1, and 4, the final array is strs = ["bc", "az"]. Both these rows are individually in lexicographic order (ie. strs[0][0] <= strs[0][1] and strs[1][0] <= strs[1][1]). Note that strs[0] > strs[1] - the array strs is not necessarily in lexicographic order. Example 2: Input: strs = ["edcba"] Output: 4 Explanation: If we delete less than 4 columns, the only row will not be lexicographically sorted.

Example 3: Input: strs = ["ghi","def","abc"] Output: 0 Explanation: All rows are already lexicographically sorted.

Constraints:

n == strs.length 1 <= n <= 100 1 <= strs[i].length <= 100 strs[i] consists of lowercase English letters.

==

Problem Number: 1831 URL: https://leetcode.com/problems/least-operators-to-express-number Title: 964. Least Operators to Express Number Problem Description: Given a single positive integer x, we will write an expression of the form x (op1) x (op2) x (op3) x ... where each operator op1, op2, etc. is either addition, subtraction, multiplication, or division (+, -, *, or /). For example, with x = 3, we might write 3 * 3 / 3 + 3 - 3 which is a value of 3. When writing such an expression, we adhere to the following conventions:

The division operator (/) returns rational numbers. There are no parentheses placed anywhere. We use the usual order of operations: multiplication and division happen before addition and subtraction. It is not allowed to use the unary negation operator (-). For example, "x - x" is a valid expression as it only uses subtraction, but "-x + x" is not because it uses negation.

We would like to write an expression with the least number of operators such that the expression equals the given target. Return the least number of operators used. Example 1: Input: x = 3, target = 19 Output: 5 Explanation: 3 * 3 + 3 * 3 + 3 / 3. The expression contains 5 operations.

Example 2: Input: x = 5, target = 501 Output: 8 Explanation: 5 * 5 * 5 * 5 - 5 * 5 * 5 + 5 / 5. The expression contains 8 operations.

Example 3: Input: x = 100, target = 100000000 Output: 3 Explanation: 100 * 100 * 100 * 100. The expression contains 3 operations.

Constraints:

2 <= x <= 100 1 <= target <= 2 * 108

==

Problem Number: 1832 URL: https://leetcode.com/problems/binary-tree-cameras Title: 968. Binary Tree Cameras Problem Description: You are given the root of a binary tree. We install cameras on the tree nodes where each camera at a node can monitor its parent, itself, and its immediate children. Return the minimum number of cameras needed to monitor all nodes of the tree. Example 1:

Input: root = [0,0,null,0,0] Output: 1 Explanation: One camera is enough to monitor all nodes if placed as shown.

Example 2:

Input: root = [0,0,null,0,null,0,null,null,0] Output: 2 Explanation: At least two cameras are needed to monitor all nodes of the tree. The above image shows one of the valid configurations of camera placement.

Constraints:

The number of nodes in the tree is in the range [1, 1000]. Node.val == 0

==

Problem Number: 1833 URL: https://leetcode.com/problems/equal-rational-numbers Title: 972. Equal Rational Numbers Problem Description: Given two strings s and t, each of which represents a non-negative rational number, return true if and only if they represent the same number. The strings may use parentheses to denote the repeating part of the rational number. A rational number can be represented using up to three parts: <IntegerPart>, <NonRepeatingPart>, and a <RepeatingPart>. The number will be represented in one of the following three ways:

<IntegerPart>

For example, 12, 0, and 123.

<IntegerPart><.><NonRepeatingPart>

For example, 0.5, 1., 2.12, and 123.0001.

<IntegerPart><.><NonRepeatingPart><(><RepeatingPart><)>

For example, 0.1(6), 1.(9), 123.00(1212).

The repeating portion of a decimal expansion is conventionally denoted within a pair of round brackets. For example:

1/6 = 0.16666666... = 0.1(6) = 0.1666(6) = 0.166(66).

Example 1: Input: s = "0.(52)", t = "0.5(25)" Output: true Explanation: Because "0.(52)" represents 0.52525252..., and "0.5(25)" represents 0.52525252525..... , the strings represent the same number.

Example 2: Input: s = "0.1666(6)", t = "0.166(66)" Output: true

Example 3: Input: s = "0.9(9)", t = "1." Output: true Explanation: "0.9(9)" represents 0.999999999... repeated forever, which equals 1. [See this link for an explanation.] "1." represents the number 1, which is formed correctly: (IntegerPart) = "1" and (NonRepeatingPart) = "".

Constraints:

Each part consists only of digits. The <IntegerPart> does not have leading zeros (except for the zero itself). 1 <= <IntegerPart>.length <= 4 0 <= <NonRepeatingPart>.length <= 4 1 <= <RepeatingPart>.length <= 4

==

Problem Number: 1834 URL: https://leetcode.com/problems/odd-even-jump

Title: 975. Odd Even Jump Problem Description: You are given an integer array arr. From some starting index, you can make a series of jumps. The (1st, 3rd, 5th, ...) jumps in the series are called odd-numbered jumps, and the (2nd, 4th, 6th, ...) jumps in the series are called even-numbered jumps. Note that the jumps are numbered, not the indices. You may jump forward from index i to index j (with i < j) in the following way:

During odd-numbered jumps (i.e., jumps 1, 3, 5, ...), you jump to the index j such that arr[i] <= arr[j] and arr[j] is the smallest possible value. If there are multiple such indices j, you can only jump to the smallest such index j. During even-numbered jumps (i.e., jumps 2, 4, 6, ...), you jump to the index j such that arr[i] >= arr[j] and arr[j] is the largest possible value. If there are multiple such indices j, you can only jump to the smallest such index j. It may be the case that for some index i, there are no legal jumps.

A starting index is good if, starting from that index, you can reach the end of the array (index arr.length - 1) by jumping some number of times (possibly 0 or more than once). Return the number of good starting indices. Example 1: Input: arr = [10,13,12,14,15] Output: 2 Explanation: From starting index i = 0, we can make our 1st jump to i = 2 (since arr[2] is the smallest among arr[1], arr[2], arr[3], arr[4] that is greater or equal to arr[0]), then we cannot jump any more. From starting index i = 1 and i = 2, we can make our 1st jump to i = 3, then we cannot jump any more. From starting index i = 3, we can make our 1st jump to i = 4, so we have reached the end. From starting index i = 4, we have reached the end already. In total, there are 2 different starting indices i = 3 and i = 4, where we can reach the end with some number of jumps.

Example 2: Input: arr = [2,3,1,1,4] Output: 3 Explanation: From starting index i = 0, we make jumps to i = 1, i = 2, i = 3: During our 1st jump (odd-numbered), we first jump to i = 1 because arr[1] is the smallest value in [arr[1], arr[2], arr[3], arr[4]] that is greater than or equal to arr[0]. During our 2nd jump (even-numbered), we jump from i = 1 to i = 2 because arr[2] is the largest value in [arr[2], arr[3], arr[4]] that is less than or equal to arr[1]. arr[3] is also the largest value, but 2 is a smaller index, so we can only jump to i = 2 and not i = 3 During our 3rd jump (odd-numbered), we jump from i = 2 to i = 3 because arr[3] is the smallest value in [arr[3], arr[4]] that is greater than or equal to arr[2]. We can't jump from i = 3 to i = 4, so the starting index i = 0 is not good. In a similar manner, we can deduce that: From starting index i = 1, we jump to i = 4, so we reach the end. From starting index i = 2, we jump to i = 3, and then we can't jump anymore. From starting index i = 3, we jump to i = 4, so we reach the end. From starting index i = 4, we are already at the end. In total, there are 3 different starting indices i = 1, i = 3, and i = 4, where we can reach the end with some number of jumps.

Example 3: Input: arr = [5,1,3,4,2] Output: 3 Explanation: We can reach the end from starting indices 1, 2, and 4.

Constraints:

1 <= arr.length <= 2 * 104 0 <= arr[i] < 105

==
Problem Number: 1835 URL: https://leetcode.com/problems/unique-paths-iii Title: 980. Unique Paths III Problem Description: You are given an m x n integer array grid where grid[i][j] could be:

1 representing the starting square. There is exactly one starting square. 2 representing the ending square. There is exactly one ending square. 0 representing empty squares we can walk over. -1 representing obstacles that we cannot walk over.

Return the number of 4-directional walks from the starting square to the ending square, that walk over every non-obstacle square exactly once.   Example 1:

Input: grid = [[1,0,0,0],[0,0,0,0],[0,0,2,-1]] Output: 2 Explanation: We have the following two paths: 1. (0,0),(0,1),(0,2),(0,3),(1,3),(1,2),(1,1),(1,0),(2,0),(2,1),(2,2) 2. (0,0),(1,0),(2,0),(2,1),(1,1),(0,1),(0,2),(0,3),(1,3),(1,2),(2,2)

Example 2:

Input: grid = [[1,0,0,0],[0,0,0,0],[0,0,0,2]] Output: 4 Explanation: We have the following four paths: 1. (0,0),(0,1),(0,2),(0,3),(1,3),(1,2),(1,1),(1,0),(2,0),(2,1),(2,2),(2,3) 2. (0,0),(0,1),(1,1),(1,0),(2,0),(2,1),(2,2),(1,2),(0,2),(0,3),(1,3),(2,3) 3. (0,0),(1,0),(2,0),(2,1),(2,2),(1,2),(1,1),(0,1 4. (0,0),(1,0),(2,0),(2,1),(1,1),(0,1),(0,2),(0,3),(1,3),(1,2),(2,2),(2,3)

Example 3:

Input: grid = [[0,1],[2,0]] Output: 0 Explanation: There is no path that walks over every empty square exactly once. Note that the starting and ending square can be anywhere in the grid.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 20 1 <= m * n <= 20 -1 <= grid[i][j] <= 2 There is exactly one starting cell and one ending cell.

==
Problem Number: 1836 URL: https://leetcode.com/problems/triples-with-bitwise-and-equal-to-zero Title: 982. Triples with Bitwise AND Equal To Zero Problem Description: Given an integer array nums, return the number of AND triples. An AND triple is a triple of indices (i, j, k) such that:

0 <= i < nums.length 0 <= j < nums.length 0 <= k < nums.length nums[i] & nums[j] & nums[k] == 0, where & represents the bitwise-AND operator.

Example 1: Input: nums = [2,1,3] Output: 12 Explanation: We could choose the following i, j, k triples: (i=0, j=0, k=1) : 2 & 2 & 1 (i=0, j=1, k=0) : 2 & 1 & 2 (i=0, j=1, k=1) : 2 & 1 & 1 (i=0, j=1, k=2) : 2 & 1 & 3 (i=0, j=2, k=1) : 2 & 3 & 1 (i=1, j=0, k=0) : 1 & 2 & 2 (i=1, j=0, k=1) : 1 & 2 & 1 (i=1, j=0, k=2) : 1 & 2 & 3 (i=1, j=1, k=0) : 1 & 1 & 2 (i=1, j=2, k=0) : 1 & 3 & 2 (i=2, j=0, k=1) : 3 & 2 & 1 (i=2, j=1, k=0) : 3 & 1 & 2

Example 2: Input: nums = [0,0,0] Output: 27

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] < 216

==
Problem Number: 1837 URL: https://leetcode.com/problems/vertical-order-traversal-of-a-binary-tree Title: 987. Vertical Order Traversal of a Binary Tree Problem Description: Given the root of a binary tree, calculate the vertical order traversal of the binary tree. For each node at position (row, col), its left and right children will be at positions (row + 1, col - 1) and (row + 1, col + 1) respectively. The root of the tree is at (0, 0). The vertical order traversal of a binary tree is a list of top-to-bottom orderings for each column index starting from the leftmost column and ending on the rightmost column. There may be multiple nodes in the same row and same column. In such a case, sort these nodes by their values. Return the vertical order traversal of the binary tree. Example 1:

Input: root = [3,9,20,null,null,15,7] Output: [[9],[3,15],[20],[7]] Explanation: Column -1: Only node 9 is in this column. Column 0: Nodes 3 and 15 are in this column in that order from top to bottom. Column 1: Only node 20 is in this column. Column 2: Only node 7 is in this column. Example 2:

Input: root = [1,2,3,4,5,6,7] Output: [[4],[2],[1,5,6],[3],[7]] Explanation: Column -2: Only node 4 is in this column. Column -1: Only node 2 is in this column. Column 0: Nodes 1, 5, and 6 are in this column. 1 is at the top, so it comes first. 5 and 6 are at the same position (2, 0), so we order them by their value, 5 before 6. Column 1: Only node 3 is in this column. Column 2: Only node 7 is in this column.

Example 3:

Input: root = [1,2,3,4,6,5,7] Output: [[4],[2],[1,5,6],[3],[7]] Explanation: This case is the exact same as example 2, but with nodes 5 and 6 swapped. Note that the solution remains the same since 5 and 6 are in the same location and should be ordered by their values.

Constraints:

The number of nodes in the tree is in the range [1, 1000]. 0 <= Node.val <= 1000

==
Problem Number: 1838 URL: https://leetcode.com/problems/subarrays-with-k-different-integers Title: 992. Subarrays with K Different Integers Problem Description: Given an integer array nums and an integer k, return the number of good subarrays of nums. A good array is an array where the number of different integers in that array is exactly k.

For example, [1,2,3,1,2] has 3 different integers: 1, 2, and 3.

A subarray is a contiguous part of an array. Example 1: Input: nums = [1,2,1,2,3], k = 2 Output: 7 Explanation: Subarrays formed with exactly 2 different integers: [1,2], [2,1], [1,2], [2,3], [1,2,1], [2,1,2], [1,2,1,2]

Example 2: Input: nums = [1,2,1,3,4], k = 3 Output: 3 Explanation: Subarrays formed with exactly 3 different integers: [1,2,1,3], [2,1,3], [1,3,4].

Constraints:

1 <= nums.length <= 2 * 104 1 <= nums[i], k <= nums.length

==
Problem Number: 1839 URL: https://leetcode.com/problems/minimum-number-of-k-consecutive-bit-flips Title: 995. Minimum Number of K Consecutive Bit Flips Problem Description: You are given a binary array nums and an integer k. A k-bit flip is choosing a subarray of length k from nums and simultaneously changing every 0 in the subarray to 1, and every 1 in the subarray to 0. Return the minimum number of k-bit flips required so that there is no 0 in the array. If it is not possible, return -1. A subarray is a contiguous part of an array. Example 1: Input: nums = [0,1,0], k = 1 Output: 2 Explanation: Flip nums[0], then flip nums[2].

Example 2: Input: nums = [1,1,0], k = 2 Output: -1 Explanation: No matter how we flip subarrays of size 2, we cannot make the array become [1,1,1].

Example 3: Input: nums = [0,0,0,1,0,1,1,0], k = 3 Output: 3 Explanation: Flip nums[0],nums[1],nums[2]: nums becomes [1,1,1,1,0,1,1,0] Flip nums[4],nums[5],nums[6]: nums becomes [1,1,1,1,1,0,0,0] Flip nums[5],nums[6],nums[7]: nums becomes [1,1,1,1,1,1,1,1]

Constraints:

1 <= nums.length <= 105 1 <= k <= nums.length

==
Problem Number: 1840 URL: https://leetcode.com/problems/number-of-squareful-arrays Title: 996. Number of Squareful Arrays Problem Description: An array is squareful if the sum of every pair of adjacent elements is a perfect square. Given an integer array nums, return the number of permutations of nums that are squareful. Two permutations perm1 and perm2 are different if there is some index i such that perm1[i] != perm2[i]. Example 1: Input: nums = [1,17,8] Output: 2 Explanation: [1,8,17] and [17,8,1] are the valid permutations.

Example 2: Input: nums = [2,2,2] Output: 1

Constraints:

1 <= nums.length <= 12 0 <= nums[i] <= 109

==
Problem Number: 1841 URL: https://leetcode.com/problems/minimum-cost-

to-merge-stones Title: 1000. Minimum Cost to Merge Stones Problem Description: There are n piles of stones arranged in a row. The ith pile has stones[i] stones. A move consists of merging exactly k consecutive piles into one pile, and the cost of this move is equal to the total number of stones in these k piles. Return the minimum cost to merge all piles of stones into one pile. If it is impossible, return -1. Example 1: Input: stones = [3,2,4,1], k = 2 Output: 20 Explanation: We start with [3, 2, 4, 1]. We merge [3, 2] for a cost of 5, and we are left with [5, 4, 1]. We merge [4, 1] for a cost of 5, and we are left with [5, 5]. We merge [5, 5] for a cost of 10, and we are left with [10]. The total cost was 20, and this is the minimum possible.

Example 2: Input: stones = [3,2,4,1], k = 3 Output: -1 Explanation: After any merge operation, there are 2 piles left, and we can't merge anymore. So the task is impossible.

Example 3: Input: stones = [3,5,1,2,6], k = 3 Output: 25 Explanation: We start with [3, 5, 1, 2, 6]. We merge [5, 1, 2] for a cost of 8, and we are left with [3, 8, 6]. We merge [3, 8, 6] for a cost of 17, and we are left with [17]. The total cost was 25, and this is the minimum possible.

Constraints:

n == stones.length 1 <= n <= 30 1 <= stones[i] <= 100 2 <= k <= 30

==
Problem Number: 1842 URL: https://leetcode.com/problems/grid-illumination Title: 1001. Grid Illumination Problem Description: There is a 2D grid of size n x n where each cell of this grid has a lamp that is initially turned off. You are given a 2D array of lamp positions lamps, where lamps[i] = [rowi, coli] indicates that the lamp at grid[rowi][coli] is turned on. Even if the same lamp is listed more than once, it is turned on. When a lamp is turned on, it illuminates its cell and all other cells in the same row, column, or diagonal. You are also given another 2D array queries, where queries[j] = [rowj, colj]. For the jth query, determine whether grid[rowj][colj] is illuminated or not. After answering the jth query, turn off the lamp at grid[rowj][colj] and its 8 adjacent lamps if they exist. A lamp is adjacent if its cell shares either a side or corner with grid[rowj][colj]. Return an array of integers ans, where ans[j] should be 1 if the cell in the jth query was illuminated, or 0 if the lamp was not. Example 1:

Input: n = 5, lamps = [[0,0],[4,4]], queries = [[1,1],[1,0]] Output: [1,0] Explanation: We have the initial grid with all lamps turned off. In the above picture we see the grid after turning on the lamp at grid[0][0] then turning on the lamp at grid[4][4]. The 0th query asks if the lamp at grid[1][1] is illuminated or not (the blue square). It is illuminated, so set ans[0] = 1. Then, we turn off all lamps in the red square.

The 1st query asks if the lamp at grid[1][0] is illuminated or not (the blue square). It is not illuminated, so set ans[1] = 0. Then, we turn off all lamps in the red rectangle.

Example 2: Input: n = 5, lamps = [[0,0],[4,4]], queries = [[1,1],[1,1]] Output: [1,1]

Example 3: Input: n = 5, lamps = [[0,0],[0,4]], queries = [[0,4],[0,1],[1,4]] Output: [1,1,0]

Constraints:

1 <= n <= 109 0 <= lamps.length <= 20000 0 <= queries.length <= 20000 lamps[i].length == 2 0 <= rowi, coli < n queries[j].length == 2 0 <= rowj, colj < n

==

Problem Number: 1843 URL: https://leetcode.com/problems/numbers-with-repeated-digits Title: 1012. Numbers With Repeated Digits Problem Description: Given an integer n, return the number of positive integers in the range [1, n] that have at least one repeated digit. Example 1: Input: n = 20 Output: 1 Explanation: The only positive number (<= 20) with at least 1 repeated digit is 11.

Example 2: Input: n = 100 Output: 10 Explanation: The positive numbers (<= 100) with atleast 1 repeated digit are 11, 22, 33, 44, 55, 66, 77, 88, 99, and 100.

Example 3: Input: n = 1000 Output: 262

Constraints:

1 <= n <= 109

==

Problem Number: 1844 URL: https://leetcode.com/problems/recover-a-tree-from-preorder-traversal Title: 1028. Recover a Tree From Preorder Traversal Problem Description: We run a preorder depth-first search (DFS) on the root of a binary tree. At each node in this traversal, we output D dashes (where D is the depth of this node), then we output the value of this node. If the depth of a node is D, the depth of its immediate child is D + 1. The depth of the root node is 0. If a node has only one child, that child is guaranteed to be the left child. Given the output traversal of this traversal, recover the tree and return its root. Example 1:

Input: traversal = "1-2--3--4-5--6--7" Output: [1,2,5,3,4,6,7]

Example 2:

Input: traversal = "1-2--3---4-5--6---7" Output: [1,2,5,3,null,6,null,4,null,7]

Example 3:

Input: traversal = "1-401--349---90--88" Output: [1,401,null,349,88,90]

Constraints:

The number of nodes in the original tree is in the range [1, 1000]. 1 <= Node.val <= 109

== Problem Number: 1845 URL: https://leetcode.com/problems/stream-of-characters Title: 1032. Stream of Characters Problem Description: Design an algorithm that accepts a stream of characters and checks if a suffix of these characters is a string of a given array of strings words. For example, if words = ["abc", "xyz"] and the stream added the four characters (one by one) 'a', 'x', 'y', and 'z', your algorithm should detect that the suffix "xyz" of the characters "axyz" matches "xyz" from words. Implement the StreamChecker class:

StreamChecker(String[] words) Initializes the object with the strings array words. boolean query(char letter) Accepts a new character from the stream and returns true if any non-empty suffix from the stream forms a word that is in words.

Example 1: Input ["StreamChecker", "query", "query", "query", "query", "query", "query", "query", "query", "query", "query", "query", "query"] [[["cd", "f", "kl"]], ["a"], ["b"], ["c"], ["d"], ["e"], ["f"], ["g"], ["h"], ["i"], ["j"], ["k"], ["l"]] Output [null, false, false, false, true, false, true, false, false, false, false, false, true]

Explanation StreamChecker streamChecker = new StreamChecker(["cd", "f", "kl"]); streamChecker.query("a"); // return False streamChecker.query("b"); // return False streamChecker.query("c"); // return False streamChecker.query("d"); // return True, because 'cd' is in the wordlist streamChecker.query("e"); // return False streamChecker.query("f"); // return True, because 'f' is in the wordlist streamChecker.query("g"); // return False streamChecker.query("h"); // return False streamChecker.query("i"); // return False streamChecker.query("j"); // return False streamChecker.query("k"); // return False streamChecker.query("l"); // return True, because 'kl' is in the wordlist

Constraints:

1 <= words.length <= 2000 1 <= words[i].length <= 200 words[i] consists of lowercase English letters. letter is a lowercase English letter. At most 4 * 104 calls will be made to query.

== Problem Number: 1846 URL: https://leetcode.com/problems/escape-a-large-maze Title: 1036. Escape a Large Maze Problem Description: There is a 1 million by 1 million grid on an XY-plane, and the coordinates of each grid square are (x, y). We start at the source = [sx, sy] square and want to reach the target = [tx, ty] square. There is also an array of blocked squares, where each blocked[i] = [xi, yi] represents a blocked square with coordinates (xi, yi). Each move, we can walk one square north, east, south, or west if the square is not in the array of blocked squares. We are also not allowed to walk outside of the grid. Return true if and only if it is possible to reach the target square from the source square through a sequence of valid moves. Example 1: Input: blocked = [[0,1],[1,0]], source = [0,0], target = [0,2] Output: false Explanation: The target square is inaccessible starting from the source square because we cannot move. We cannot move north or east because those squares are blocked. We cannot move south or west because we cannot go outside of the grid.

Example 2: Input: blocked = [], source = [0,0], target = [999999,999999] Output: true Explanation: Because there are no blocked cells, it is possible to reach the target square.

Constraints:

0 <= blocked.length <= 200 blocked[i].length == 2 0 <= xi, yi < 106 source.length == target.length == 2 0 <= sx, sy, tx, ty < 106 source != target It is guaranteed that source and target are not blocked.

==

Problem Number: 1847 URL: https://leetcode.com/problems/longest-duplicate-substring Title: 1044. Longest Duplicate Substring Problem Description: Given a string s, consider all duplicated substrings: (contiguous) substrings of s that occur 2 or more times. The occurrences may overlap. Return any duplicated substring that has the longest possible length. If s does not have a duplicated substring, the answer is "". Example 1: Input: s = "banana" Output: "ana" Example 2: Input: s = "abcd" Output: ""

Constraints:

2 <= s.length <= 3 * 104 s consists of lowercase English letters.

==

Problem Number: 1848 URL: https://leetcode.com/problems/number-of-submatrices-that-sum-to-target Title: 1074. Number of Submatrices That Sum to Target Problem Description: Given a matrix and a target, return the number of non-empty submatrices that sum to target. A submatrix x1, y1, x2, y2 is the set of all cells matrix[x][y] with x1 <= x <= x2 and y1 <= y <= y2. Two submatrices (x1, y1, x2, y2) and (x1', y1', x2', y2') are different if they have some coordinate that is different: for example, if x1 != x1'. Example 1:

Input: matrix = [[0,1,0],[1,1,1],[0,1,0]], target = 0 Output: 4 Explanation: The four 1x1 submatrices that only contain 0.

Example 2: Input: matrix = [[1,-1],[-1,1]], target = 0 Output: 5 Explanation: The two 1x2 submatrices, plus the two 2x1 submatrices, plus the 2x2 submatrix.

Example 3: Input: matrix = [[904]], target = 0 Output: 0

Constraints:

1 <= matrix.length <= 100 1 <= matrix[0].length <= 100 -1000 <= matrix[i] <= 1000 -10^8 <= target <= 10^8

==
Problem Number: 1849 URL: https://leetcode.com/problems/shortest-common-supersequence Title: 1092. Shortest Common Supersequence Problem Description: Given two strings str1 and str2, return the shortest string that has both str1 and str2 as subsequences. If there are multiple valid strings, return any of them. A string s is a subsequence of string t if deleting some number of characters from t (possibly 0) results in the string s. Example 1: Input: str1 = "abac", str2 = "cab" Output: "cabac" Explanation: str1 = "abac" is a subsequence of "cabac" because we can delete the first "c". str2 = "cab" is a subsequence of "cabac" because we can delete the last "ac". The answer provided is the shortest such string that satisfies these properties.

Example 2: Input: str1 = "aaaaaaaa", str2 = "aaaaaaaa" Output: "aaaaaaaa"

Constraints:

1 <= str1.length, str2.length <= 1000 str1 and str2 consist of lowercase English letters.

==
Problem Number: 1850 URL: https://leetcode.com/problems/find-in-mountain-array Title: 1095. Find in Mountain Array Problem Description: (This problem is an interactive problem.) You may recall that an array arr is a mountain array if and only if:

arr.length >= 3 There exists some i with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i] arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Given a mountain array mountainArr, return the minimum index such that mountainArr.get(index) == target. If such an index does not exist, return -1. You cannot access the mountain array directly. You may only access the array using a MountainArray interface:

MountainArray.get(k) returns the element of the array at index k (0-indexed). MountainArray.length() returns the length of the array.

Submissions making more than 100 calls to MountainArray.get will be judged Wrong Answer. Also, any solutions that attempt to circumvent the judge will result in disqualification. Example 1: Input: array = [1,2,3,4,5,3,1], target = 3 Output: 2 Explanation: 3 exists in the array, at index=2 and index=5. Return the minimum index, which is 2. Example 2: Input: array = [0,1,2,4,2,1], target = 3 Output: -1 Explanation: 3 does not exist in the array, so we return -1.

Constraints:

3 <= mountain_arr.length() <= 104 0 <= target <= 109 0 <= mountain_arr.get(index) <= 109

==
Problem Number: 1851 URL: https://leetcode.com/problems/brace-expansion-

ii Title: 1096. Brace Expansion II Problem Description: Under the grammar given below, strings can represent a set of lowercase words. Let R(expr) denote the set of words the expression represents. The grammar can best be understood through simple examples:

Single letters represent a singleton set containing that word.

R(”a”) = {”a”} R(”w”) = {”w”}

When we take a comma-delimited list of two or more expressions, we take the union of possibilities.

R(”{a,b,c}”) = {”a”,”b”,”c”} R(”{{a,b},{b,c}}”) = {”a”,”b”,”c”} (notice the final set only contains each word at most once)

When we concatenate two expressions, we take the set of possible concatenations between two words where the first word comes from the first expression and the second word comes from the second expression.

R(”{a,b}{c,d}”) = {”ac”,”ad”,”bc”,”bd”} R(”a{b,c}{d,e}f{g,h}”) = {”abdfg”, ”abdfh”, ”abefg”, ”abefh”, ”acdfg”, ”acdfh”, ”acefg”, ”acefh”}

Formally, the three rules for our grammar:

For every lowercase letter x, we have R(x) = {x}. For expressions e1, e2, ... , ek with k >= 2, we have R({e1, e2, ...}) = R(e1) R(e2) ... For expressions e1 and e2, we have R(e1 + e2) = {a + b for (a, b) in R(e1) × R(e2)}, where + denotes concatenation, and × denotes the cartesian product.

Given an expression representing a set of words under the given grammar, return the sorted list of words that the expression represents. Example 1: Input: expression = ”{a,b}{c,{d,e}}” Output: [”ac”,”ad”,”ae”,”bc”,”bd”,”be”]

Example 2: Input: expression = ”{{a,z},a{b,c},{ab,z}}” Output: [”a”,”ab”,”ac”,”z”] Explanation: Each distinct word is written only once in the final answer.

Constraints:

1 <= expression.length <= 60 expression[i] consists of '{', '}', ','or lowercase English letters. The given expression represents a set of words based on the grammar given in the description.

==

Problem Number: 1852 URL: https://leetcode.com/problems/parsing-a-boolean-expression Title: 1106. Parsing A Boolean Expression Problem Description: A boolean expression is an expression that evaluates to either true or false. It can be in one of the following shapes:

't' that evaluates to true. 'f' that evaluates to false. '!(subExpr)' that evaluates to the logical NOT of the inner expression subExpr. '&(subExpr1, subExpr2, ..., subExprn)' that evaluates to the logical AND of the inner expressions subExpr1, subExpr2, ..., subExprn where n >= 1. '|(subExpr1, subExpr2, ..., subExprn)' that evaluates to the logical OR of the inner expressions subExpr1, subExpr2, ..., subExprn where n >= 1.

Given a string expression that represents a boolean expression, return the evaluation of that expression. It is guaranteed that the given expression is valid and follows the given rules. Example 1: Input: expression = ”&(|(f))” Output: false Explanation: First, evaluate |(f) --> f. The expression is now ”&(f)”. Then, evaluate &(f) --> f. The expression is now ”f”. Finally, return false.

Example 2: Input: expression = ”|(f,f,f,t)” Output: true Explanation: The evaluation of (false OR false OR false OR true) is true.

Example 3: Input: expression = ”!(&(f,t))” Output: true Explanation: First, evaluate &(f,t) --> (false AND true) --> false --> f. The expression is now ”!(f)”. Then, evaluate !(f) --> NOT false --> true. We return true.

Constraints:

1 <= expression.length <= 2 * 104 expression[i] is one following characters: ’(’, ’)’, ’&’, ’|’, ’!’, ’t’, ’f’, and ’,’.

==
Problem Number: 1853 URL: https://leetcode.com/problems/smallest-sufficient-team Title: 1125. Smallest Sufficient Team Problem Description: In a project, you have a list of required skills req_skills, and a list of people. The ith person people[i] contains a list of skills that the person has. Consider a sufficient team: a set of people such that for every required skill in req_skills, there is at least one person in the team who has that skill. We can represent these teams by the index of each person.

For example, team = [0, 1, 3] represents the people with skills people[0], people[1], and people[3].

Return any sufficient team of the smallest possible size, represented by the index of each person. You may return the answer in any order. It is guaranteed an answer exists. Example 1: Input: req_skills = [”java”,”nodejs”,”reactjs”], people = [[”java”],[”nodejs”],[”nodejs”,”reactjs”]] Output: [0,2] Example 2: Input: req_skills = [”algorithms”,”math”,”java”,”reactjs”,”csharp”,”aws”], people = [[”algorithms”,”math”,”java”],[”algorithms”,”math”,”reactjs”],[”java”,”csharp”,”aws”],[”reactjs”,”csharp”],[”cs
Output: [1,2]

Constraints:

1 <= req_skills.length <= 16 1 <= req_skills[i].length <= 16 req_skills[i] consists of lowercase English letters. All the strings of req_skills are unique. 1 <= people.length <= 60 0 <= people[i].length <= 16 1 <= people[i][j].length <= 16 people[i][j] consists of lowercase English letters. All the strings of people[i] are unique. Every skill in people[i] is a skill in req_skills. It is guaranteed a sufficient team exists.

==

Problem Number: 1854 URL: https://leetcode.com/problems/longest-chunked-palindrome-decomposition Title: 1147. Longest Chunked Palindrome Decomposition Problem Description: You are given a string text. You should split it to k substrings (subtext1, subtext2, ..., subtextk) such that:

subtexti is a non-empty string. The concatenation of all the substrings is equal to text (i.e., subtext1 + subtext2 + ... + subtextk == text). subtexti == subtextk - i + 1 for all valid values of i (i.e., 1 <= i <= k).

Return the largest possible value of k. Example 1: Input: text = "ghiabcdefhelloadamhelloabcdefghi" Output: 7 Explanation: We can split the string on "(ghi)(abcdef)(hello)(adam)(hello)(abcdef)(ghi)".

Example 2: Input: text = "merchant" Output: 1 Explanation: We can split the string on "(merchant)".

Example 3: Input: text = "antaprezatepzapreanta" Output: 11 Explanation: We can split the string on "(a)(nt)(a)(pre)(za)(tep)(za)(pre)(a)(nt)(a)".

Constraints:

1 <= text.length <= 1000 text consists only of lowercase English characters.

==
Problem Number: 1855 URL: https://leetcode.com/problems/online-majority-element-in-subarray Title: 1157. Online Majority Element In Subarray Problem Description: Design a data structure that efficiently finds the majority element of a given subarray. The majority element of a subarray is an element that occurs threshold times or more in the subarray. Implementing the MajorityChecker class:

MajorityChecker(int[] arr) Initializes the instance of the class with the given array arr. int query(int left, int right, int threshold) returns the element in the subarray arr[left...right] that occurs at least threshold times, or -1 if no such element exists.

Example 1: Input ["MajorityChecker", "query", "query", "query"] [[[1, 1, 2, 2, 1, 1]], [0, 5, 4], [0, 3, 3], [2, 3, 2]] Output [null, 1, -1, 2]

Explanation MajorityChecker majorityChecker = new MajorityChecker([1, 1, 2, 2, 1, 1]); majorityChecker.query(0, 5, 4); // return 1 majorityChecker.query(0, 3, 3); // return -1 majorityChecker.query(2, 3, 2); // return 2

Constraints:

1 <= arr.length <= 2 * 104 1 <= arr[i] <= 2 * 104 0 <= left <= right < arr.length threshold <= right - left + 1 2 * threshold > right - left + 1 At most 104 calls will be made to query.

==
Problem Number: 1856 URL: https://leetcode.com/problems/last-substring-in-lexicographical-order Title: 1163. Last Substring in Lexicographical Order

Problem Description: Given a string s, return the last substring of s in lexicographical order. Example 1: Input: s = ”abab” Output: ”bab” Explanation: The substrings are [”a”, ”ab”, ”aba”, ”abab”, ”b”, ”ba”, ”bab”]. The lexicographically maximum substring is ”bab”.

Example 2: Input: s = ”leetcode” Output: ”tcode”

Constraints:

1 <= s.length <= 4 * 105 s contains only lowercase English letters.

==

Problem Number: 1857 URL: https://leetcode.com/problems/dinner-plate-stacks Title: 1172. Dinner Plate Stacks Problem Description: You have an infinite number of stacks arranged in a row and numbered (left to right) from 0, each of the stacks has the same maximum capacity. Implement the DinnerPlates class:

DinnerPlates(int capacity) Initializes the object with the maximum capacity of the stacks capacity. void push(int val) Pushes the given integer val into the leftmost stack with a size less than capacity. int pop() Returns the value at the top of the rightmost non-empty stack and removes it from that stack, and returns -1 if all the stacks are empty. int popAtStack(int index) Returns the value at the top of the stack with the given index index and removes it from that stack or returns -1 if the stack with that given index is empty.

Example 1: Input [”DinnerPlates”, ”push”, ”push”, ”push”, ”push”, ”push”, ”popAtStack”, ”push”, ”push”, ”popAtStack”, ”popAtStack”, ”pop”, ”pop”, ”pop”, ”pop”, ”pop”] [[2], [1], [2], [3], [4], [5], [0], [20], [21], [0], [2], [], [], [], [], []] Output [null, null, null, null, null, null, 2, null, null, 20, 21, 5, 4, 3, 1, -1]

Explanation: DinnerPlates D = DinnerPlates(2); // Initialize with capacity = 2 D.push(1); D.push(2); D.push(3); D.push(4); D.push(5); // The stacks are now: 2 4 1 3 5 D.popAtStack(0); // Returns 2. The stacks are now: 4 1 3 5 D.push(20); // The stacks are now: 20 4 1 3 5 D.push(21); // The stacks are now: 20 4 21 1 3 5 D.popAtStack(0); // Returns 20. The stacks are now: 4 21 1 3 5 D.popAtStack(2); // Returns 21. The stacks are now: 4 1 3 5 D.pop() // Returns 5. The stacks are now: 4 1 3 D.pop() // Returns 4. The stacks are now: 1 3 D.pop() // Returns 3. The stacks are now: 1 D.pop() // Returns 1. There are no stacks. D.pop() // Returns -1. There are still no stacks.

Constraints:

1 <= capacity <= 2 * 104 1 <= val <= 2 * 104 0 <= index <= 105 At most 2 * 105 calls will be made to push, pop, and popAtStack.

==

Problem Number: 1858 URL: https://leetcode.com/problems/number-of-valid-words-for-each-puzzle Title: 1178. Number of Valid Words for Each Puzzle

Problem Description: With respect to a given puzzle string, a word is valid if both the following conditions are satisfied:

word contains the first letter of puzzle. For each letter in word, that letter is in puzzle.

For example, if the puzzle is "abcdefg", then valid words are "faced", "cabbage", and "baggage", while invalid words are "beefed" (does not include 'a') and "based" (includes 's' which is not in the puzzle).

Return an array answer, where answer[i] is the number of words in the given word list words that is valid with respect to the puzzle puzzles[i]. Example 1: Input: words = ["aaaa","asas","able","ability","actt","actor","access"], puzzles = ["aboveyz","abrodyz","abslute","absoryz","actresz","gaswxyz"] Output: [1,1,3,2,4,0] Explanation: 1 valid word for "aboveyz" : "aaaa" 1 valid word for "abrodyz" : "aaaa" 3 valid words for "abslute" : "aaaa", "asas", "able" 2 valid words for "absoryz" : "aaaa", "asas" 4 valid words for "actresz" : "aaaa", "asas", "actt", "access" There are no valid words for "gaswxyz" cause none of the words in the list contains letter 'g'.

Example 2: Input: words = ["apple","pleas","please"], puzzles = ["aelwxyz","aelpxyz","aelpsxy","saelpxy","xaelpsy"] Output: [0,1,3,2,0]

Constraints:

1 <= words.length <= 105 4 <= words[i].length <= 50 1 <= puzzles.length <= 104 puzzles[i].length == 7 words[i] and puzzles[i] consist of lowercase English letters. Each puzzles[i] does not contain repeated characters.

==
Problem Number: 1859 URL: https://leetcode.com/problems/make-array-strictly-increasing Title: 1187. Make Array Strictly Increasing Problem Description: Given two integer arrays arr1 and arr2, return the minimum number of operations (possibly zero) needed to make arr1 strictly increasing. In one operation, you can choose two indices 0 <= i < arr1.length and 0 <= j < arr2.length and do the assignment arr1[i] = arr2[j]. If there is no way to make arr1 strictly increasing, return -1. Example 1: Input: arr1 = [1,5,3,6,7], arr2 = [1,3,2,4] Output: 1 Explanation: Replace 5 with 2, then arr1 = [1, 2, 3, 6, 7].

Example 2: Input: arr1 = [1,5,3,6,7], arr2 = [4,3,1] Output: 2 Explanation: Replace 5 with 3 and then replace 3 with 4. arr1 = [1, 3, 4, 6, 7].

Example 3: Input: arr1 = [1,5,3,6,7], arr2 = [1,6,3,3] Output: -1 Explanation: You can't make arr1 strictly increasing. Constraints:

1 <= arr1.length, arr2.length <= 2000 0 <= arr1[i], arr2[i] <= 10^9

==
Problem Number: 1860 URL: https://leetcode.com/problems/critical-

connections-in-a-network Title: 1192. Critical Connections in a Network Problem Description: There are n servers numbered from 0 to n - 1 connected by undirected server-to-server connections forming a network where connections[i] = [ai, bi] represents a connection between servers ai and bi. Any server can reach other servers directly or indirectly through the network. A critical connection is a connection that, if removed, will make some servers unable to reach some other server. Return all critical connections in the network in any order. Example 1:

Input: n = 4, connections = [[0,1],[1,2],[2,0],[1,3]] Output: [[1,3]] Explanation: [[3,1]] is also accepted.

Example 2: Input: n = 2, connections = [[0,1]] Output: [[0,1]]

Constraints:

2 <= n <= 105 n - 1 <= connections.length <= 105 0 <= ai, bi <= n - 1 ai != bi There are no repeated connections.

==
Problem Number: 1861 URL: https://leetcode.com/problems/sort-items-by-groups-respecting-dependencies Title: 1203. Sort Items by Groups Respecting Dependencies Problem Description: There are n items each belonging to zero or one of m groups where group[i] is the group that the i-th item belongs to and it's equal to -1 if the i-th item belongs to no group. The items and the groups are zero indexed. A group can have no item belonging to it. Return a sorted list of the items such that:

The items that belong to the same group are next to each other in the sorted list. There are some relations between these items where beforeItems[i] is a list containing all the items that should come before the i-th item in the sorted array (to the left of the i-th item).

Return any solution if there is more than one solution and return an empty list if there is no solution. Example 1:

Input: n = 8, m = 2, group = [-1,-1,1,0,0,1,0,-1], beforeItems = [[],[6],[5],[6],[3,6],[],[],[]] Output: [6,3,4,1,5,2,0,7]

Example 2: Input: n = 8, m = 2, group = [-1,-1,1,0,0,1,0,-1], beforeItems = [[],[6],[5],[6],[3],[],[4],[]] Output: [] Explanation: This is the same as example 1 except that 4 needs to be before 6 in the sorted list.

Constraints:

1 <= m <= n <= 3 * 104 group.length == beforeItems.length == n -1 <= group[i] <= m - 1 0 <= beforeItems[i].length <= n - 1 0 <= beforeItems[i][j] <= n - 1 i != beforeItems[i][j] beforeItems[i] does not contain duplicates elements.

==
Problem Number: 1862 URL: https://leetcode.com/problems/design-skiplist Title: 1206. Design Skiplist Problem Description: Design a Skiplist without

using any built-in libraries. A skiplist is a data structure that takes O(log(n)) time to add, erase and search. Comparing with treap and red-black tree which has the same function and performance, the code length of Skiplist can be comparatively short and the idea behind Skiplists is just simple linked lists. For example, we have a Skiplist containing [30,40,50,60,70,90] and we want to add 80 and 45 into it. The Skiplist works this way:

Artyom Kalinin [CC BY-SA 3.0], via Wikimedia Commons You can see there are many layers in the Skiplist. Each layer is a sorted linked list. With the help of the top layers, add, erase and search can be faster than O(n). It can be proven that the average time complexity for each operation is O(log(n)) and space complexity is O(n). See more about Skiplist: https://en.wikipedia.org/wiki/Skip_list Implement the Skiplist class:

Skiplist() Initializes the object of the skiplist. bool search(int target) Returns true if the integer target exists in the Skiplist or false otherwise. void add(int num) Inserts the value num into the SkipList. bool erase(int num) Removes the value num from the Skiplist and returns true. If num does not exist in the Skiplist, do nothing and return false. If there exist multiple num values, removing any one of them is fine.

Note that duplicates may exist in the Skiplist, your code needs to handle this situation. Example 1: Input ["Skiplist", "add", "add", "add", "search", "add", "search", "erase", "erase", "search"] [[], [1], [2], [3], [0], [4], [1], [0], [1], [1]] Output [null, null, null, null, false, null, true, false, true, false]

Explanation Skiplist skiplist = new Skiplist(); skiplist.add(1); skiplist.add(2); skiplist.add(3); skiplist.search(0); // return False skiplist.add(4); skiplist.search(1); // return True skiplist.erase(0); // return False, 0 is not in skiplist. skiplist.erase(1); // return True skiplist.search(1); // return False, 1 has already been erased. Constraints:

0 <= num, target <= 2 * 104 At most 5 * 104 calls will be made to search, add, and erase.

==
Problem Number: 1863 URL: https://leetcode.com/problems/minimum-moves-to-reach-target-with-rotations Title: 1210. Minimum Moves to Reach Target with Rotations Problem Description: In an n*n grid, there is a snake that spans 2 cells and starts moving from the top left corner at (0, 0) and (0, 1). The grid has empty cells represented by zeros and blocked cells represented by ones. The snake wants to reach the lower right corner at (n-1, n-2) and (n-1, n-1). In one move the snake can:

Move one cell to the right if there are no blocked cells there. This move keeps the horizontal/vertical position of the snake as it is. Move down one cell if there are no blocked cells there. This move keeps the horizontal/vertical position of the snake as it is. Rotate clockwise if it's in a horizontal position and the two cells under it are both empty. In that case the snake moves from (r, c) and (r, c+1) to (r, c) and (r+1, c).

Rotate counterclockwise if it's in a vertical position and the two cells to its right are both empty. In that case the snake moves from (r, c) and (r+1, c) to (r, c) and (r, c+1).

Return the minimum number of moves to reach the target. If there is no way to reach the target, return -1. Example 1:

Input: grid = [[0,0,0,0,0,1], [1,1,0,0,1,0], [0,0,0,0,1,1], [0,0,1,0,1,0], [0,1,1,0,0,0], [0,1,1,0,0,0]] Output: 11 Explanation: One possible solution is [right, right, rotate clockwise, right, down, down, down, down, rotate counterclockwise, right, down].

Example 2: Input: grid = [[0,0,1,1,1,1], [0,0,0,0,1,1], [1,1,0,0,0,1], [1,1,1,0,0,1], [1,1,1,0,0,1], [1,1,1,0,0,0]] Output: 9

Constraints:

2 <= n <= 100 0 <= grid[i][j] <= 1 It is guaranteed that the snake starts at empty cells.

==
Problem Number: 1864 URL: https://leetcode.com/problems/count-vowels-permutation Title: 1220. Count Vowels Permutation Problem Description: Given an integer n, your task is to count how many strings of length n can be formed under the following rules:

Each character is a lower case vowel ('a', 'e', 'i', 'o', 'u') Each vowel 'a' may only be followed by an 'e'. Each vowel 'e' may only be followed by an 'a' or an 'i'. Each vowel 'i' may not be followed by another 'i'. Each vowel 'o' may only be followed by an 'i' or a 'u'. Each vowel 'u' may only be followed by an 'a'.

Since the answer may be too large, return it modulo 10^9 + 7. Example 1: Input: n = 1 Output: 5 Explanation: All possible strings are: "a", "e", "i" , "o" and "u".

Example 2: Input: n = 2 Output: 10 Explanation: All possible strings are: "ae", "ea", "ei", "ia", "ie", "io", "iu", "oi", "ou" and "ua".

Example 3: Input: n = 5 Output: 68 Constraints:

1 <= n <= 2 * 10^4

==
Problem Number: 1865 URL: https://leetcode.com/problems/dice-roll-simulation Title: 1223. Dice Roll Simulation Problem Description: A die simulator generates a random number from 1 to 6 for each roll. You introduced a constraint to the generator such that it cannot roll the number i more than rollMax[i] (1-indexed) consecutive times. Given an array of integers rollMax and an integer n, return the number of distinct sequences that can be obtained with exact n rolls. Since the answer may be too large, return it modulo 109 +

7. Two sequences are considered different if at least one element differs from each other. Example 1: Input: n = 2, rollMax = [1,1,2,2,2,3] Output: 34 Explanation: There will be 2 rolls of die, if there are no constraints on the die, there are 6 * 6 = 36 possible combinations. In this case, looking at rollMax array, the numbers 1 and 2 appear at most once consecutively, therefore sequences (1,1) and (2,2) cannot occur, so the final answer is 36-2 = 34.

Example 2: Input: n = 2, rollMax = [1,1,1,1,1,1] Output: 30

Example 3: Input: n = 3, rollMax = [1,1,1,2,2,3] Output: 181

Constraints:

1 <= n <= 5000 rollMax.length == 6 1 <= rollMax[i] <= 15

==

Problem Number: 1866 URL: https://leetcode.com/problems/maximum-equal-frequency Title: 1224. Maximum Equal Frequency Problem Description: Given an array nums of positive integers, return the longest possible length of an array prefix of nums, such that it is possible to remove exactly one element from this prefix so that every number that has appeared in it will have the same number of occurrences. If after removing one element there are no remaining elements, it's still considered that every appeared number has the same number of ocurrences (0). Example 1: Input: nums = [2,2,1,1,5,3,3,5] Output: 7 Explanation: For the subarray [2,2,1,1,5,3,3] of length 7, if we remove nums[4] = 5, we will get [2,2,1,1,3,3], so that each number will appear exactly twice.

Example 2: Input: nums = [1,1,1,2,2,2,3,3,3,4,4,4,5] Output: 13

Constraints:

2 <= nums.length <= 105 1 <= nums[i] <= 105

==

Problem Number: 1867 URL: https://leetcode.com/problems/maximum-profit-in-job-scheduling Title: 1235. Maximum Profit in Job Scheduling Problem Description: We have n jobs, where every job is scheduled to be done from startTime[i] to endTime[i], obtaining a profit of profit[i]. You're given the startTime, endTime and profit arrays, return the maximum profit you can take such that there are no two jobs in the subset with overlapping time range. If you choose a job that ends at time X you will be able to start another job that starts at time X. Example 1:

Input: startTime = [1,2,3,3], endTime = [3,4,5,6], profit = [50,10,40,70] Output: 120 Explanation: The subset chosen is the first and fourth job. Time range [1-3]+[3-6] , we get profit of 120 = 50 + 70.

Example 2:

Input: startTime = [1,2,3,4,6], endTime = [3,5,10,6,9], profit = [20,20,100,70,60] Output: 150 Explanation: The subset chosen is the first, fourth and fifth job. Profit obtained 150 = 20 + 70 + 60.

Example 3:

Input: startTime = [1,1,1], endTime = [2,3,4], profit = [5,6,4] Output: 6

Constraints:

1 <= startTime.length == endTime.length == profit.length <= 5 * 104 1 <= startTime[i] < endTime[i] <= 109 1 <= profit[i] <= 104

==
Problem Number: 1868 URL: https://leetcode.com/problems/tiling-a-rectangle-with-the-fewest-squares Title: 1240. Tiling a Rectangle with the Fewest Squares Problem Description: Given a rectangle of size n x m, return the minimum number of integer-sided squares that tile the rectangle. Example 1:

Input: n = 2, m = 3 Output: 3 Explanation: 3 squares are necessary to cover the rectangle. 2 (squares of 1x1) 1 (square of 2x2) Example 2:

Input: n = 5, m = 8 Output: 5

Example 3:

Input: n = 11, m = 13 Output: 6

Constraints:

1 <= n, m <= 13

==
Problem Number: 1869 URL: https://leetcode.com/problems/check-if-it-is-a-good-array Title: 1250. Check If It Is a Good Array Problem Description: Given an array nums of positive integers. Your task is to select some subset of nums, multiply each element by an integer and add all these numbers. The array is said to be good if you can obtain a sum of 1 from the array by any possible subset and multiplicand. Return True if the array is good otherwise return False. Example 1: Input: nums = [12,5,7,23] Output: true Explanation: Pick numbers 5 and 7. 5*3 + 7*(-2) = 1

Example 2: Input: nums = [29,6,10] Output: true Explanation: Pick numbers 29, 6 and 10. 29*1 + 6*(-3) + 10*(-1) = 1

Example 3: Input: nums = [3,6] Output: false

Constraints:

1 <= nums.length <= 10^5 1 <= nums[i] <= 10^9

==
Problem Number: 1870 URL: https://leetcode.com/problems/maximum-score-words-formed-by-letters Title: 1255. Maximum Score Words Formed by Letters Problem Description: Given a list of words, list of single letters (might be repeating) and score of every character. Return the maximum score of any valid set of words formed by using the given letters (words[i] cannot be used two or more times). It is not necessary to use all characters in letters and

each letter can only be used once. Score of letters ’a’, ’b’, ’c’, ... ,’z’ is given by score[0], score[1], ... , score[25] respectively. Example 1: Input: words = [”dog”,”cat”,”dad”,”good”], letters = [”a”,”a”,”c”,”d”,”d”,”d”,”g”,”o”,”o”], score = [1,0,9,5,0,0,3,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0] Output: 23 Explanation: Score a=1, c=9, d=5, g=3, o=2 Given letters, we can form the words ”dad” (5+1+5) and ”good” (3+2+2+5) with a score of 23. Words ”dad” and ”dog” only get a score of 21. Example 2: Input: words = [”xxxz”,”ax”,”bx”,”cx”], letters = [”z”,”a”,”b”,”c”,”x”,”x”,”x”], score = [4,4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,10] Output: 27 Explanation: Score a=4, b=4, c=4, x=5, z=10 Given letters, we can form the words ”ax” (4+5), ”bx” (4+5) and ”cx” (4+5) with a score of 27. Word ”xxxz” only get a score of 25. Example 3: Input: words = [”leetcode”], letters = [”l”,”e”,”t”,”c”,”o”,”d”], score = [0,0,1,1,1,0,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,0,0,0,0,0] Output: 0 Explanation: Letter ”e” can only be used once. Constraints:

1 <= words.length <= 14 1 <= words[i].length <= 15 1 <= letters.length <= 100 letters[i].length == 1 score.length == 26 0 <= score[i] <= 10 words[i], letters[i] contains only lower case English letters.

==
Problem Number: 1871 URL: https://leetcode.com/problems/minimum-moves-to-move-a-box-to-their-target-location Title: 1263. Minimum Moves to Move a Box to Their Target Location Problem Description: A storekeeper is a game in which the player pushes boxes around in a warehouse trying to get them to target locations. The game is represented by an m x n grid of characters grid where each element is a wall, floor, or box. Your task is to move the box ’B’ to the target position ’T’ under the following rules:

The character ’S’ represents the player. The player can move up, down, left, right in grid if it is a floor (empty cell). The character ’.’ represents the floor which means a free cell to walk. The character ’#’ represents the wall which means an obstacle (impossible to walk there). There is only one box ’B’ and one target cell ’T’ in the grid. The box can be moved to an adjacent free cell by standing next to the box and then moving in the direction of the box. This is a push. The player cannot walk through the box.

Return the minimum number of pushes to move the box to the target. If there is no way to reach the target, return -1. Example 1:

Input: grid = [[”#”,”#”,”#”,”#”,”#”,”#”], [”#”,”T”,”#”,”#”,”#”,”#”], [”#”,”.”,”.”,”B”,”.”,”#”], [”#”,”.”,”#”,”#”,”.”,”#”], [”#”,”.”,”.”,”.”,”S”,”#”], [”#”,”#”,”#”,”#”,”#”,”#”]] Output: 3 Explanation: We return only the number of times the box is pushed. Example 2: Input: grid = [[”#”,”#”,”#”,”#”,”#”,”#”], [”#”,”T”,”#”,”#”,”#”,”#”], [”#”,”.”,”.”,”B”,”.”,”#”], [”#”,”#”,”#”,”#”,”.”,”#”], [”#”,”.”,”.”,”.”,”S”,”#”], [”#”,”#”,”#”,”#”,”#”,”#”]] Output: -1

Example 3: Input: grid = [[”#”,”#”,”#”,”#”,”#”,”#”], [”#”,”T”,”.”,”.”,”#”,”#”], [”#”,”.”,”#”,”B”,”.”,”#”], [”#”,”.”,”.”,”.”,”.”,”#”], [”#”,”.”,”.”,”.”,”S”,”#”],

["#","#","#","#","#","#"]] Output: 5 Explanation: push the box down, left, left, up and up.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 20 grid contains only characters '.', '#', 'S', 'T', or 'B'. There is only one character 'S', 'B', and 'T' in the grid.

===
Problem Number: 1872 URL: https://leetcode.com/problems/number-of-ways-to-stay-in-the-same-place-after-some-steps Title: 1269. Number of Ways to Stay in the Same Place After Some Steps Problem Description: You have a pointer at index 0 in an array of size arrLen. At each step, you can move 1 position to the left, 1 position to the right in the array, or stay in the same place (The pointer should not be placed outside the array at any time). Given two integers steps and arrLen, return the number of ways such that your pointer is still at index 0 after exactly steps steps. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: steps = 3, arrLen = 2 Output: 4 Explanation: There are 4 differents ways to stay at index 0 after 3 steps. Right, Left, Stay Stay, Right, Left Right, Stay, Left Stay, Stay, Stay

Example 2: Input: steps = 2, arrLen = 4 Output: 2 Explanation: There are 2 differents ways to stay at index 0 after 2 steps Right, Left Stay, Stay

Example 3: Input: steps = 4, arrLen = 2 Output: 8

Constraints:

1 <= steps <= 500 1 <= arrLen <= 106

===
Problem Number: 1873 URL: https://leetcode.com/problems/palindrome-partitioning-iii Title: 1278. Palindrome Partitioning III Problem Description: You are given a string s containing lowercase letters and an integer k. You need to :

First, change some characters of s to other lowercase English letters. Then divide s into k non-empty disjoint substrings such that each substring is a palindrome.

Return the minimal number of characters that you need to change to divide the string. Example 1: Input: s = "abc", k = 2 Output: 1 Explanation: You can split the string into "ab" and "c", and change 1 character in "ab" to make it palindrome.

Example 2: Input: s = "aabbc", k = 3 Output: 0 Explanation: You can split the string into "aa", "bb" and "c", all of them are palindrome. Example 3: Input: s = "leetcode", k = 8 Output: 0

Constraints:

1 <= k <= s.length <= 100. s only contains lowercase English letters.

==
Problem Number: 1874 URL: https://leetcode.com/problems/minimum-number-of-flips-to-convert-binary-matrix-to-zero-matrix Title: 1284. Minimum Number of Flips to Convert Binary Matrix to Zero Matrix Problem Description: Given a m x n binary matrix mat. In one step, you can choose one cell and flip it and all the four neighbors of it if they exist (Flip is changing 1 to 0 and 0 to 1). A pair of cells are called neighbors if they share one edge. Return the minimum number of steps required to convert mat to a zero matrix or -1 if you cannot. A binary matrix is a matrix with all cells equal to 0 or 1 only. A zero matrix is a matrix with all cells equal to 0. Example 1:

Input: mat = [[0,0],[0,1]] Output: 3 Explanation: One possible solution is to flip (1, 0) then (0, 1) and finally (1, 1) as shown.

Example 2: Input: mat = [[0]] Output: 0 Explanation: Given matrix is a zero matrix. We do not need to change it.

Example 3: Input: mat = [[1,0,0],[1,0,0]] Output: -1 Explanation: Given matrix cannot be a zero matrix.

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 3 mat[i][j] is either 0 or 1.

==
Problem Number: 1875 URL: https://leetcode.com/problems/minimum-falling-path-sum-ii Title: 1289. Minimum Falling Path Sum II Problem Description: Given an n x n integer matrix grid, return the minimum sum of a falling path with non-zero shifts. A falling path with non-zero shifts is a choice of exactly one element from each row of grid such that no two elements chosen in adjacent rows are in the same column. Example 1:

Input: grid = [[1,2,3],[4,5,6],[7,8,9]] Output: 13 Explanation: The possible falling paths are: [1,5,9], [1,5,7], [1,6,7], [1,6,8], [2,4,8], [2,4,9], [2,6,7], [2,6,8], [3,4,8], [3,4,9], [3,5,7], [3,5,9] The falling path with the smallest sum is [1,5,7], so the answer is 13.

Example 2: Input: grid = [[7]] Output: 7

Constraints:

n == grid.length == grid[i].length 1 <= n <= 200 -99 <= grid[i][j] <= 99

==
Problem Number: 1876 URL: https://leetcode.com/problems/shortest-path-in-a-grid-with-obstacles-elimination Title: 1293. Shortest Path in a Grid with Obstacles Elimination Problem Description: You are given an m x n integer matrix grid where each cell is either 0 (empty) or 1 (obstacle). You can move up, down, left, or right from and to an empty cell in one step. Return the minimum number of steps to walk from the upper left corner (0, 0) to the lower

right corner (m - 1, n - 1) given that you can eliminate at most k obstacles. If it is not possible to find such walk return -1.   Example 1:

Input: grid = [[0,0,0],[1,1,0],[0,0,0],[0,1,1],[0,0,0]], k = 1 Output: 6 Explanation: The shortest path without eliminating any obstacle is 10. The shortest path with one obstacle elimination at position (3,2) is 6. Such path is (0,0) -> (0,1) -> (0,2) -> (1,2) -> (2,2) -> (3,2) -> (4,2).

Example 2:

Input: grid = [[0,1,1],[1,1,1],[1,0,0]], k = 1 Output: -1 Explanation: We need to eliminate at least two obstacles to find such a walk.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 40 1 <= k <= m * n grid[i][j] is either 0 or 1. grid[0][0] == grid[m - 1][n - 1] == 0

==

Problem Number: 1877 URL: https://leetcode.com/problems/maximum-candies-you-can-get-from-boxes Title: 1298. Maximum Candies You Can Get from Boxes Problem Description: You have n boxes labeled from 0 to n - 1. You are given four arrays: status, candies, keys, and containedBoxes where:

status[i] is 1 if the ith box is open and 0 if the ith box is closed, candies[i] is the number of candies in the ith box, keys[i] is a list of the labels of the boxes you can open after opening the ith box. containedBoxes[i] is a list of the boxes you found inside the ith box.

You are given an integer array initialBoxes that contains the labels of the boxes you initially have. You can take all the candies in any open box and you can use the keys in it to open new boxes and you also can use the boxes you find in it. Return the maximum number of candies you can get following the rules above. Example 1: Input: status = [1,0,1,0], candies = [7,5,4,100], keys = [[],[],[1],[]], containedBoxes = [[1,2],[3],[],[]], initialBoxes = [0] Output: 16 Explanation: You will be initially given box 0. You will find 7 candies in it and boxes 1 and 2. Box 1 is closed and you do not have a key for it so you will open box 2. You will find 4 candies and a key to box 1 in box 2. In box 1, you will find 5 candies and box 3 but you will not find a key to box 3 so box 3 will remain closed. Total number of candies collected = 7 + 4 + 5 = 16 candy.

Example 2: Input: status = [1,0,0,0,0,0], candies = [1,1,1,1,1,1], keys = [[1,2,3,4,5],[],[],[],[],[]], containedBoxes = [[1,2,3,4,5],[],[],[],[],[]], initialBoxes = [0] Output: 6 Explanation: You have initially box 0. Opening it you can find boxes 1,2,3,4 and 5 and their keys. The total number of candies will be 6.

Constraints:

n == status.length == candies.length == keys.length == containedBoxes.length 1 <= n <= 1000 status[i] is either 0 or 1. 1 <= candies[i] <= 1000 0 <= keys[i].length <= n 0 <= keys[i][j] < n All values of keys[i] are unique. 0 <= containedBoxes[i].length <= n 0 <= containedBoxes[i][j] < n All values of containedBoxes[i] are unique. Each box is contained in one box at most. 0 <= initialBoxes.length <= n 0 <= initialBoxes[i] < n

==
Problem Number: 1878 URL: https://leetcode.com/problems/number-of-paths-with-max-score Title: 1301. Number of Paths with Max Score Problem Description: You are given a square board of characters. You can move on the board starting at the bottom right square marked with the character 'S'. You need to reach the top left square marked with the character 'E'. The rest of the squares are labeled either with a numeric character 1, 2, ..., 9 or with an obstacle 'X'. In one move you can go up, left or up-left (diagonally) only if there is no obstacle there. Return a list of two integers: the first integer is the maximum sum of numeric characters you can collect, and the second is the number of such paths that you can take to get that maximum sum, taken modulo 10^9 + 7. In case there is no path, return [0, 0]. Example 1: Input: board = ["E23","2X2","12S"] Output: [7,1] Example 2: Input: board = ["E12","1X1","21S"] Output: [4,2] Example 3: Input: board = ["E11","XXX","11S"] Output: [0,0]

Constraints:

2 <= board.length == board[i].length <= 100

==
Problem Number: 1879 URL: https://leetcode.com/problems/verbal-arithmetic-puzzle Title: 1307. Verbal Arithmetic Puzzle Problem Description: Given an equation, represented by words on the left side and the result on the right side. You need to check if the equation is solvable under the following rules:

Each character is decoded as one digit (0 - 9). No two characters can map to the same digit. Each words[i] and result are decoded as one number without leading zeros. Sum of numbers on the left side (words) will equal to the number on the right side (result).

Return true if the equation is solvable, otherwise return false. Example 1: Input: words = ["SEND","MORE"], result = "MONEY" Output: true Explanation: Map 'S'-> 9, 'E'->5, 'N'->6, 'D'->7, 'M'->1, 'O'->0, 'R'->8, 'Y'->'2' Such that: "SEND" + "MORE" = "MONEY" , 9567 + 1085 = 10652 Example 2: Input: words = ["SIX","SEVEN","SEVEN"], result = "TWENTY" Output: true Explanation: Map 'S'-> 6, 'I'->5, 'X'->0, 'E'->8, 'V'->7, 'N'->2, 'T'->1, 'W'->'3', 'Y'->4 Such that: "SIX" + "SEVEN" + "SEVEN" = "TWENTY" , 650 + 68782 + 68782 = 138214 Example 3: Input: words = ["LEET","CODE"], result = "POINT" Output: false Explanation: There is no possible mapping to satisfy the equation, so we return false. Note that two different characters cannot map to the same digit.

Constraints:

2 <= words.length <= 5 1 <= words[i].length, result.length <= 7 words[i], result contain only uppercase English letters. The number of different characters used in the expression is at most 10.

==
Problem Number: 1880 URL: https://leetcode.com/problems/minimum-insertion-steps-to-make-a-string-palindrome Title: 1312. Minimum Insertion Steps to Make a String Palindrome Problem Description: Given a string s. In one step you can insert any character at any index of the string. Return the minimum number of steps to make s palindrome. A Palindrome String is one that reads the same backward as well as forward. Example 1: Input: s = "zzazz" Output: 0 Explanation: The string "zzazz" is already palindrome we do not need any insertions.

Example 2: Input: s = "mbadm" Output: 2 Explanation: String can be "mbdadbm" or "mdbabdm".

Example 3: Input: s = "leetcode" Output: 5 Explanation: Inserting 5 characters the string becomes "leetcodocteel".

Constraints:

1 <= s.length <= 500 s consists of lowercase English letters.

==
Problem Number: 1881 URL: https://leetcode.com/problems/distinct-echo-substrings Title: 1316. Distinct Echo Substrings Problem Description: Return the number of distinct non-empty substrings of text that can be written as the concatenation of some string with itself (i.e. it can be written as a + a where a is some string). Example 1: Input: text = "abcabcabc" Output: 3 Explanation: The 3 substrings are "abcabc", "bcabca" and "cabcab".

Example 2: Input: text = "leetcodeleetcode" Output: 2 Explanation: The 2 substrings are "ee" and "leetcodeleetcode".

Constraints:

1 <= text.length <= 2000 text has only lowercase English letters.

==
Problem Number: 1882 URL: https://leetcode.com/problems/minimum-distance-to-type-a-word-using-two-fingers Title: 1320. Minimum Distance to Type a Word Using Two Fingers Problem Description:

You have a keyboard layout as shown above in the X-Y plane, where each English uppercase letter is located at some coordinate.

For example, the letter 'A' is located at coordinate (0, 0), the letter 'B' is located at coordinate (0, 1), the letter 'P' is located at coordinate (2, 3) and the letter 'Z' is located at coordinate (4, 1).

Given the string word, return the minimum total distance to type such string using only two fingers. The distance between coordinates (x1, y1) and (x2, y2) is |x1 - x2| + |y1 - y2|. Note that the initial positions of your two fingers are considered free so do not count towards your total distance, also your two fingers do not have to start at the first letter or the first two letters. Example 1: Input: word = ”CAKE” Output: 3 Explanation: Using two fingers, one optimal way to type ”CAKE” is: Finger 1 on letter ’C’ -> cost = 0 Finger 1 on letter ’A’ -> cost = Distance from letter ’C’ to letter ’A’ = 2 Finger 2 on letter ’K’ -> cost = 0 Finger 2 on letter ’E’ -> cost = Distance from letter ’K’ to letter ’E’ = 1 Total distance = 3

Example 2: Input: word = ”HAPPY” Output: 6 Explanation: Using two fingers, one optimal way to type ”HAPPY” is: Finger 1 on letter ’H’ -> cost = 0 Finger 1 on letter ’A’ -> cost = Distance from letter ’H’ to letter ’A’ = 2 Finger 2 on letter ’P’ -> cost = 0 Finger 2 on letter ’P’ -> cost = Distance from letter ’P’ to letter ’P’ = 0 Finger 1 on letter ’Y’ -> cost = Distance from letter ’A’ to letter ’Y’ = 4 Total distance = 6

Constraints:

2 <= word.length <= 300 word consists of uppercase English letters.

==

Problem Number: 1883 URL: https://leetcode.com/problems/minimum-number-of-taps-to-open-to-water-a-garden Title: 1326. Minimum Number of Taps to Open to Water a Garden Problem Description: There is a one-dimensional garden on the x-axis. The garden starts at the point 0 and ends at the point n. (i.e., the length of the garden is n). There are n + 1 taps located at points [0, 1, ..., n] in the garden. Given an integer n and an integer array ranges of length n + 1 where ranges[i] (0-indexed) means the i-th tap can water the area [i - ranges[i], i + ranges[i]] if it was open. Return the minimum number of taps that should be open to water the whole garden, If the garden cannot be watered return -1. Example 1:

Input: n = 5, ranges = [3,4,1,1,0,0] Output: 1 Explanation: The tap at point 0 can cover the interval [-3,3] The tap at point 1 can cover the interval [-3,5] The tap at point 2 can cover the interval [1,3] The tap at point 3 can cover the interval [2,4] The tap at point 4 can cover the interval [4,4] The tap at point 5 can cover the interval [5,5] Opening Only the second tap will water the whole garden [0,5]

Example 2: Input: n = 3, ranges = [0,0,0,0] Output: -1 Explanation: Even if you activate all the four taps you cannot water the whole garden.

Constraints:

1 <= n <= 104 ranges.length == n + 1 0 <= ranges[i] <= 100

==

Problem Number: 1884 URL: https://leetcode.com/problems/reverse-subarray-to-maximize-array-value Title: 1330. Reverse Subarray To Maximize Array Value Problem Description: You are given an integer array nums. The value of this array is defined as the sum of |nums[i] - nums[i + 1]| for all 0 <= i < nums.length - 1. You are allowed to select any subarray of the given array and reverse it. You can perform this operation only once. Find maximum possible value of the final array.   Example 1: Input: nums = [2,3,1,5,4] Output: 10 Explanation: By reversing the subarray [3,1,5] the array becomes [2,5,1,3,4] whose value is 10.

Example 2: Input: nums = [2,4,9,24,2,1,10] Output: 68

Constraints:

1 <= nums.length <= 3 * 104 -105 <= nums[i] <= 105

==
Problem Number: 1885 URL: https://leetcode.com/problems/minimum-difficulty-of-a-job-schedule Title: 1335. Minimum Difficulty of a Job Schedule Problem Description: You want to schedule a list of jobs in d days. Jobs are dependent (i.e To work on the ith job, you have to finish all the jobs j where 0 <= j < i). You have to finish at least one task every day. The difficulty of a job schedule is the sum of difficulties of each day of the d days. The difficulty of a day is the maximum difficulty of a job done on that day. You are given an integer array jobDifficulty and an integer d. The difficulty of the ith job is jobDifficulty[i]. Return the minimum difficulty of a job schedule. If you cannot find a schedule for the jobs return -1.   Example 1:

Input: jobDifficulty = [6,5,4,3,2,1], d = 2 Output: 7 Explanation: First day you can finish the first 5 jobs, total difficulty = 6. Second day you can finish the last job, total difficulty = 1. The difficulty of the schedule = 6 + 1 = 7

Example 2: Input: jobDifficulty = [9,9,9], d = 4 Output: -1 Explanation: If you finish a job per day you will still have a free day. you cannot find a schedule for the given jobs.

Example 3: Input: jobDifficulty = [1,1,1], d = 3 Output: 3 Explanation: The schedule is one job per day. total difficulty will be 3.

Constraints:

1 <= jobDifficulty.length <= 300 0 <= jobDifficulty[i] <= 1000 1 <= d <= 10

==
Problem Number: 1886 URL: https://leetcode.com/problems/jump-game-v Title: 1340. Jump Game V Problem Description: Given an array of integers arr and an integer d. In one step you can jump from index i to index:

i + x where: i + x < arr.length and 0 < x <= d. i - x where: i - x >= 0 and 0 < x <= d.

In addition, you can only jump from index i to index j if arr[i] > arr[j] and arr[i] > arr[k] for all indices k between i and j (More formally min(i, j) < k <

max(i, j)). You can choose any index of the array and start jumping. Return the maximum number of indices you can visit. Notice that you can not jump outside of the array at any time.   Example 1:

Input: arr = [6,4,14,6,8,13,9,7,10,6,12], d = 2 Output: 4 Explanation: You can start at index 10. You can jump 10 --> 8 --> 6 --> 7 as shown. Note that if you start at index 6 you can only jump to index 7. You cannot jump to index 5 because 13 > 9. You cannot jump to index 4 because index 5 is between index 4 and 6 and 13 > 9. Similarly You cannot jump from index 3 to index 2 or index 1.

Example 2: Input: arr = [3,3,3,3,3], d = 3 Output: 1 Explanation: You can start at any index. You always cannot jump to any index.

Example 3: Input: arr = [7,6,5,4,3,2,1], d = 1 Output: 7 Explanation: Start at index 0. You can visit all the indicies.

Constraints:

1 <= arr.length <= 1000 1 <= arr[i] <= 105 1 <= d <= arr.length

==

Problem Number: 1887 URL: https://leetcode.com/problems/jump-game-iv Title: 1345. Jump Game IV Problem Description: Given an array of integers arr, you are initially positioned at the first index of the array. In one step you can jump from index i to index:

i + 1 where: i + 1 < arr.length. i - 1 where: i - 1 >= 0. j where: arr[i] == arr[j] and i != j.

Return the minimum number of steps to reach the last index of the array. Notice that you can not jump outside of the array at any time.   Example 1: Input: arr = [100,-23,-23,404,100,23,23,23,3,404] Output: 3 Explanation: You need three jumps from index 0 --> 4 --> 3 --> 9. Note that index 9 is the last index of the array.

Example 2: Input: arr = [7] Output: 0 Explanation: Start index is the last index. You do not need to jump.

Example 3: Input: arr = [7,6,9,6,9,6,9,7] Output: 1 Explanation: You can jump directly from index 0 to index 7 which is last index of the array.

Constraints:

1 <= arr.length <= 5 * 104 -108 <= arr[i] <= 108

==

Problem Number: 1888 URL: https://leetcode.com/problems/maximum-students-taking-exam Title: 1349. Maximum Students Taking Exam Problem Description: Given a m * n matrix seats that represent seats distributions in a classroom. If a seat is broken, it is denoted by '#' character otherwise it is denoted by a '.' character. Students can see the answers of those sitting next to the left, right, upper left and upper right, but he cannot see the answers of the student sitting directly in front or behind him. Return the maximum number of students that can take the exam together without any cheating being possible.. Students must be placed in seats in good condition. Example 1:

Input: seats = [["#",".","#","#",".","#"], [".","#","#","#","#","."], ["#",".","#","#",".","#"]] Output: 4 Explanation: Teacher can place 4 students in available seats so they don't cheat on the exam.

Example 2: Input: seats = [[".","#"], ["#","#"], ["#","."], ["#","#"], [".","#"]] Output: 3 Explanation: Place all students in available seats.

Example 3: Input: seats = [["#",".",".",".","#"], [".","#",".","#","."], [".",".","#",".","."], [".","#",".","#","."], ["#",".",".",".","#"]] Output: 10 Explanation: Place students in available seats in column 1, 3 and 5.

Constraints:

seats contains only characters '.' and'#'. m == seats.length n == seats[i].length 1 <= m <= 8 1 <= n <= 8

==

Problem Number: 1889 URL: https://leetcode.com/problems/construct-target-array-with-multiple-sums Title: 1354. Construct Target Array With Multiple Sums Problem Description: You are given an array target of n integers. From a starting array arr consisting of n 1's, you may perform the following procedure :

let x be the sum of all elements currently in your array. choose index i, such that 0 <= i < n and set the value of arr at index i to x. You may repeat this procedure as many times as needed.

Return true if it is possible to construct the target array from arr, otherwise, return false. Example 1: Input: target = [9,3,5] Output: true Explanation: Start with arr = [1, 1, 1] [1, 1, 1], sum = 3 choose index 1 [1, 3, 1], sum = 5 choose index 2 [1, 3, 5], sum = 9 choose index 0 [9, 3, 5] Done

Example 2: Input: target = [1,1,1,2] Output: false Explanation: Impossible to create target array from [1,1,1,1].

Example 3: Input: target = [8,5] Output: true

Constraints:

n == target.length 1 <= n <= 5 * 104 1 <= target[i] <= 109

==

Problem Number: 1890 URL: https://leetcode.com/problems/count-all-valid-pickup-and-delivery-options Title: 1359. Count All Valid Pickup and Delivery Options Problem Description: Given n orders,each order consists of a pickup and a delivery service. Count all valid pickup/delivery possible sequences such that delivery(i) is always after of pickup(i). Since the answer may be too large, return it modulo 10^9 + 7. Example 1: Input: n = 1 Output: 1 Explanation: Unique order (P1, D1), Delivery 1 always is after of Pickup 1.

Example 2: Input: n = 2 Output: 6 Explanation: All possible orders: (P1,P2,D1,D2), (P1,P2,D2,D1), (P1,D1,P2,D2), (P2,P1,D1,D2), (P2,P1,D2,D1) and (P2,D2,P1,D1). This is an invalid order (P1,D2,P2,D1) because Pickup 2 is after of Delivery 2.

Example 3: Input: n = 3 Output: 90

Constraints:

1 <= n <= 500

==

Problem Number: 1891 URL: https://leetcode.com/problems/largest-multiple-of-three Title: 1363. Largest Multiple of Three Problem Description: Given an array of digits digits, return the largest multiple of three that can be formed by concatenating some of the given digits in any order. If there is no answer return an empty string. Since the answer may not fit in an integer data type, return the answer as a string. Note that the returning answer must not contain unnecessary leading zeros. Example 1: Input: digits = [8,1,9] Output: "981"

Example 2: Input: digits = [8,6,7,1,0] Output: "8760"

Example 3: Input: digits = [1] Output: ""

Constraints:

1 <= digits.length <= 104 0 <= digits[i] <= 9

==

Problem Number: 1892 URL: https://leetcode.com/problems/minimum-cost-to-make-at-least-one-valid-path-in-a-grid Title: 1368. Minimum Cost to Make at Least One Valid Path in a Grid Problem Description: Given an m x n grid. Each cell of the grid has a sign pointing to the next cell you should visit if you are currently in this cell. The sign of grid[i][j] can be:

1 which means go to the cell to the right. (i.e go from grid[i][j] to grid[i][j + 1]) 2 which means go to the cell to the left. (i.e go from grid[i][j] to grid[i][j - 1]) 3 which means go to the lower cell. (i.e go from grid[i][j] to grid[i + 1][j]) 4 which means go to the upper cell. (i.e go from grid[i][j] to grid[i - 1][j])

Notice that there could be some signs on the cells of the grid that point outside the grid. You will initially start at the upper left cell (0, 0). A valid path in the grid is a path that starts from the upper left cell (0, 0) and ends at the bottom-right cell (m - 1, n - 1) following the signs on the grid. The valid path does not have to be the shortest. You can modify the sign on a cell with cost = 1. You can modify the sign on a cell one time only. Return the minimum cost to make the grid have at least one valid path. Example 1:

Input: grid = [[1,1,1,1],[2,2,2,2],[1,1,1,1],[2,2,2,2]] Output: 3 Explanation: You will start at point (0, 0). The path to (3, 3) is as follows. (0, 0) --> (0, 1) --> (0, 2) --> (0, 3) change the arrow to down with cost = 1 --> (1, 3) --> (1, 2) --> (1, 1) --> (1, 0) change the arrow to down with cost = 1 --> (2, 0) --> (2, 1) --> (2, 2) --> (2, 3) change the arrow to down with cost = 1 --> (3, 3) The total cost = 3.

Example 2:

Input: grid = [[1,1,3],[3,2,2],[1,1,4]] Output: 0 Explanation: You can follow the path from (0, 0) to (2, 2).

Example 3:

Input: grid = [[1,2],[4,3]] Output: 1

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 100 1 <= grid[i][j] <= 4

==

Problem Number: 1893 URL: https://leetcode.com/problems/maximum-sum-bst-in-binary-tree Title: 1373. Maximum Sum BST in Binary Tree Problem Description: Given a binary tree root, return the maximum sum of all keys of any sub-tree which is also a Binary Search Tree (BST). Assume a BST is defined as follows:

The left subtree of a node contains only nodes with keys less than the node's key. The right subtree of a node contains only nodes with keys greater than the node's key. Both the left and right subtrees must also be binary search trees.

Example 1:

Input: root = [1,4,3,2,4,2,5,null,null,null,null,null,null,4,6] Output: 20 Explanation: Maximum sum in a valid Binary search tree is obtained in root node with key equal to 3.

Example 2:

Input: root = [4,3,null,1,2] Output: 2 Explanation: Maximum sum in a valid Binary search tree is obtained in a single root node with key equal to 2.

Example 3: Input: root = [-4,-2,-5] Output: 0 Explanation: All values are negatives. Return an empty BST.

Constraints:

The number of nodes in the tree is in the range [1, 4 * 104]. -4 * 104 <= Node.val <= 4 * 104

==

Problem Number: 1894 URL: https://leetcode.com/problems/frog-position-after-t-seconds Title: 1377. Frog Position After T Seconds Problem Description: Given an undirected tree consisting of n vertices numbered from 1 to n. A frog starts jumping from vertex 1. In one second, the frog jumps from its current vertex to another unvisited vertex if they are directly connected. The frog can not jump back to a visited vertex. In case the frog can jump to several vertices, it jumps randomly to one of them with the same probability. Otherwise, when the frog can not jump to any unvisited vertex, it jumps forever on the same vertex. The edges of the undirected tree are given in the array edges, where edges[i] = [ai, bi] means that exists an edge connecting the vertices ai and bi. Return the probability that after t seconds the frog is on the vertex target. Answers within 10-5 of the actual answer will be accepted.   Example 1:

Input: n = 7, edges = [[1,2],[1,3],[1,7],[2,4],[2,6],[3,5]], t = 2, target = 4 Output: 0.16666666666666666 Explanation: The figure above shows the given graph. The frog starts at vertex 1, jumping with 1/3 probability to the vertex 2 after second 1 and then jumping with 1/2 probability to vertex 4 after second 2. Thus the probability for the frog is on the vertex 4 after 2 seconds is 1/3 * 1/2 = 1/6 = 0.16666666666666666.

Example 2:

Input: n = 7, edges = [[1,2],[1,3],[1,7],[2,4],[2,6],[3,5]], t = 1, target = 7 Output: 0.3333333333333333 Explanation: The figure above shows the given graph. The frog starts at vertex 1, jumping with 1/3 = 0.3333333333333333 probability to the vertex 7 after second 1.

Constraints:

1 <= n <= 100 edges.length == n - 1 edges[i].length == 2 1 <= ai, bi <= n 1 <= t <= 50 1 <= target <= n

==
Problem Number: 1895 URL: https://leetcode.com/problems/maximum-performance-of-a-team Title: 1383. Maximum Performance of a Team Problem Description: You are given two integers n and k and two integer arrays speed and efficiency both of length n. There are n engineers numbered from 1 to n. speed[i] and efficiency[i] represent the speed and efficiency of the ith engineer respectively. Choose at most k different engineers out of the n engineers to form a team with the maximum performance. The performance of a team is the sum of its engineers' speeds multiplied by the minimum efficiency among its engineers. Return the maximum performance of this team. Since the answer can be a huge number, return it modulo 109 + 7.   Example 1: Input: n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 2 Output: 60 Explanation: We have the maximum performance of the team by selecting engineer 2 (with speed=10 and efficiency=4) and engineer 5 (with speed=5 and efficiency=7). That is, performance = (10 + 5) * min(4, 7) = 60.

Example 2: Input: n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 3 Output: 68 Explanation: This is the same example as the first but k = 3. We can select engineer 1, engineer 2 and engineer 5 to get the maximum performance of the team. That is, performance = (2 + 10 + 5) * min(5, 4, 7)

= 68.

Example 3: Input: n = 6, speed = [2,10,3,1,5,8], efficiency = [5,4,3,9,7,2], k = 4 Output: 72

Constraints:

1 <= k <= n <= 105 speed.length == n efficiency.length == n 1 <= speed[i] <= 105 1 <= efficiency[i] <= 108

==
Problem Number: 1896 URL: https://leetcode.com/problems/pizza-with-3n-slices Title: 1388. Pizza With 3n Slices Problem Description: There is a pizza with 3n slices of varying size, you and your friends will take slices of pizza as follows:

You will pick any pizza slice. Your friend Alice will pick the next slice in the anti-clockwise direction of your pick. Your friend Bob will pick the next slice in the clockwise direction of your pick. Repeat until there are no more slices of pizzas.

Given an integer array slices that represent the sizes of the pizza slices in a clockwise direction, return the maximum possible sum of slice sizes that you can pick. Example 1:

Input: slices = [1,2,3,4,5,6] Output: 10 Explanation: Pick pizza slice of size 4, Alice and Bob will pick slices with size 3 and 5 respectively. Then Pick slices with size 6, finally Alice and Bob will pick slice of size 2 and 1 respectively. Total = 4 + 6.

Example 2:

Input: slices = [8,9,8,6,1,1] Output: 16 Explanation: Pick pizza slice of size 8 in each turn. If you pick slice with size 9 your partners will pick slices of size 8.

Constraints:

3 * n == slices.length 1 <= slices.length <= 500 1 <= slices[i] <= 1000

==
Problem Number: 1897 URL: https://leetcode.com/problems/longest-happy-prefix Title: 1392. Longest Happy Prefix Problem Description: A string is called a happy prefix if is a non-empty prefix which is also a suffix (excluding itself). Given a string s, return the longest happy prefix of s. Return an empty string ”” if no such prefix exists. Example 1: Input: s = ”level” Output: ”l” Explanation: s contains 4 prefix excluding itself (”l”, ”le”, ”lev”, ”leve”), and suffix (”l”, ”el”, ”vel”, ”evel”). The largest prefix which is also suffix is given by ”l”.

Example 2: Input: s = ”ababab” Output: ”abab” Explanation: ”abab” is the largest prefix which is also suffix. They can overlap in the original string.

Constraints:

1 <= s.length <= 105 s contains only lowercase English letters.

==
Problem Number: 1898 URL: https://leetcode.com/problems/find-all-good-strings Title: 1397. Find All Good Strings Problem Description: Given the strings s1 and s2 of size n and the string evil, return the number of good strings. A good string has size n, it is alphabetically greater than or equal to s1, it is alphabetically smaller than or equal to s2, and it does not contain the string evil as a substring. Since the answer can be a huge number, return this modulo 109 + 7. Example 1: Input: n = 2, s1 = "aa", s2 = "da", evil = "b" Output: 51 Explanation: There are 25 good strings starting with 'a': "aa","ac","ad",...,"az". Then there are 25 good strings starting with 'c': "ca","cc","cd",...,"cz" and finally there is one good string starting with 'd': "da".

Example 2: Input: n = 8, s1 = "leetcode", s2 = "leetgoes", evil = "leet" Output: 0 Explanation: All strings greater than or equal to s1 and smaller than or equal to s2 start with the prefix "leet", therefore, there is not any good string.

Example 3: Input: n = 2, s1 = "gx", s2 = "gz", evil = "x" Output: 2

Constraints:

s1.length == n s2.length == n s1 <= s2 1 <= n <= 500 1 <= evil.length <= 50 All strings consist of lowercase English letters.

==
Problem Number: 1899 URL: https://leetcode.com/problems/reducing-dishes Title: 1402. Reducing Dishes Problem Description: A chef has collected data on the satisfaction level of his n dishes. Chef can cook any dish in 1 unit of time. Like-time coefficient of a dish is defined as the time taken to cook that dish including previous dishes multiplied by its satisfaction level i.e. time[i] * satisfaction[i]. Return the maximum sum of like-time coefficient that the chef can obtain after dishes preparation. Dishes can be prepared in any order and the chef can discard some dishes to get this maximum value. Example 1: Input: satisfaction = [-1,-8,0,5,-9] Output: 14 Explanation: After Removing the second and last dish, the maximum total like-time coefficient will be equal to (-1*1 + 0*2 + 5*3 = 14). Each dish is prepared in one unit of time. Example 2: Input: satisfaction = [4,3,2] Output: 20 Explanation: Dishes can be prepared in any order, (2*1 + 3*2 + 4*3 = 20)

Example 3: Input: satisfaction = [-1,-4,-5] Output: 0 Explanation: People do not like the dishes. No dish is prepared.

Constraints:

n == satisfaction.length 1 <= n <= 500 -1000 <= satisfaction[i] <= 1000

==
Problem Number: 1900 URL: https://leetcode.com/problems/stone-game-iii Title: 1406. Stone Game III Problem Description: Alice and Bob continue

their games with piles of stones. There are several stones arranged in a row, and each stone has an associated value which is an integer given in the array stoneValue. Alice and Bob take turns, with Alice starting first. On each player's turn, that player can take 1, 2, or 3 stones from the first remaining stones in the row. The score of each player is the sum of the values of the stones taken. The score of each player is 0 initially. The objective of the game is to end with the highest score, and the winner is the player with the highest score and there could be a tie. The game continues until all the stones have been taken. Assume Alice and Bob play optimally. Return "Alice" if Alice will win, "Bob" if Bob will win, or "Tie" if they will end the game with the same score. Example 1: Input: stoneValue = [1,2,3,7] Output: "Bob" Explanation: Alice will always lose. Her best move will be to take three piles and the score become 6. Now the score of Bob is 7 and Bob wins.

Example 2: Input: stoneValue = [1,2,3,-9] Output: "Alice" Explanation: Alice must choose all the three piles at the first move to win and leave Bob with negative score. If Alice chooses one pile her score will be 1 and the next move Bob's score becomes 5. In the next move, Alice will take the pile with value = -9 and lose. If Alice chooses two piles her score will be 3 and the next move Bob's score becomes 3. In the next move, Alice will take the pile with value = -9 and also lose. Remember that both play optimally so here Alice will choose the scenario that makes her win.

Example 3: Input: stoneValue = [1,2,3,6] Output: "Tie" Explanation: Alice cannot win this game. She can end the game in a draw if she decided to choose all the first three piles, otherwise she will lose.

Constraints:

1 <= stoneValue.length <= 5 * 104 -1000 <= stoneValue[i] <= 1000

==

Problem Number: 1901 URL: https://leetcode.com/problems/number-of-ways-to-paint-n-3-grid Title: 1411. Number of Ways to Paint N × 3 Grid Problem Description: You have a grid of size n x 3 and you want to paint each cell of the grid with exactly one of the three colors: Red, Yellow, or Green while making sure that no two adjacent cells have the same color (i.e., no two cells that share vertical or horizontal sides have the same color). Given n the number of rows of the grid, return the number of ways you can paint this grid. As the answer may grow large, the answer must be computed modulo 109 + 7. Example 1:

Input: n = 1 Output: 12 Explanation: There are 12 possible way to paint the grid as shown.

Example 2: Input: n = 5000 Output: 30228214

Constraints:

n == grid.length 1 <= n <= 5000

== Problem Number: 1902 URL: https://leetcode.com/problems/restore-the-array Title: 1416. Restore The Array Problem Description: A program was supposed to print an array of integers. The program forgot to print whitespaces and the array is printed as a string of digits s and all we know is that all integers in the array were in the range [1, k] and there are no leading zeros in the array. Given the string s and the integer k, return the number of the possible arrays that can be printed as s using the mentioned program. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: s = "1000", k = 10000 Output: 1 Explanation: The only possible array is [1000]

Example 2: Input: s = "1000", k = 10 Output: 0 Explanation: There cannot be an array that was printed this way and has all integer >= 1 and <= 10.

Example 3: Input: s = "1317", k = 2000 Output: 8 Explanation: Possible arrays are [1317],[131,7],[13,17],[1,317],[13,1,7],[1,31,7],[1,3,17],[1,3,1,7]

Constraints:

1 <= s.length <= 105 s consists of only digits and does not contain leading zeros. 1 <= k <= 109

== Problem Number: 1903 URL: https://leetcode.com/problems/build-array-where-you-can-find-the-maximum-exactly-k-comparisons Title: 1420. Build Array Where You Can Find The Maximum Exactly K Comparisons Problem Description: You are given three integers n, m and k. Consider the following algorithm to find the maximum element of an array of positive integers:

You should build the array arr which has the following properties:

arr has exactly n integers. 1 <= arr[i] <= m where (0 <= i < n). After applying the mentioned algorithm to arr, the value search_cost is equal to k.

Return the number of ways to build the array arr under the mentioned conditions. As the answer may grow large, the answer must be computed modulo 109 + 7. Example 1: Input: n = 2, m = 3, k = 1 Output: 6 Explanation: The possible arrays are [1, 1], [2, 1], [2, 2], [3, 1], [3, 2] [3, 3]

Example 2: Input: n = 5, m = 2, k = 3 Output: 0 Explanation: There are no possible arrays that satisify the mentioned conditions.

Example 3: Input: n = 9, m = 1, k = 1 Output: 1 Explanation: The only possible array is [1, 1, 1, 1, 1, 1, 1, 1, 1]

Constraints:

1 <= n <= 50 1 <= m <= 100 0 <= k <= n

== Problem Number: 1904 URL: https://leetcode.com/problems/constrained-subsequence-sum Title: 1425. Constrained Subsequence Sum Problem

Description: Given an integer array nums and an integer k, return the maximum sum of a non-empty subsequence of that array such that for every two consecutive integers in the subsequence, nums[i] and nums[j], where i < j, the condition j - i <= k is satisfied. A subsequence of an array is obtained by deleting some number of elements (can be zero) from the array, leaving the remaining elements in their original order. Example 1: Input: nums = [10,2,-10,5,20], k = 2 Output: 37 Explanation: The subsequence is [10, 2, 5, 20].

Example 2: Input: nums = [-1,-2,-3], k = 1 Output: -1 Explanation: The subsequence must be non-empty, so we choose the largest number.

Example 3: Input: nums = [10,-2,-10,-5,20], k = 2 Output: 23 Explanation: The subsequence is [10, -2, -5, 20].

Constraints:

1 <= k <= nums.length <= 105 -104 <= nums[i] <= 104

==

Problem Number: 1905 URL: https://leetcode.com/problems/number-of-ways-to-wear-different-hats-to-each-other Title: 1434. Number of Ways to Wear Different Hats to Each Other Problem Description: There are n people and 40 types of hats labeled from 1 to 40. Given a 2D integer array hats, where hats[i] is a list of all hats preferred by the ith person. Return the number of ways that the n people wear different hats to each other. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: hats = [[3,4],[4,5],[5]] Output: 1 Explanation: There is only one way to choose hats given the conditions. First person choose hat 3, Second person choose hat 4 and last one hat 5.

Example 2: Input: hats = [[3,5,1],[3,5]] Output: 4 Explanation: There are 4 ways to choose hats: (3,5), (5,3), (1,3) and (1,5)

Example 3: Input: hats = [[1,2,3,4],[1,2,3,4],[1,2,3,4],[1,2,3,4]] Output: 24 Explanation: Each person can choose hats labeled from 1 to 4. Number of Permutations of (1,2,3,4) = 24.

Constraints:

n == hats.length 1 <= n <= 10 1 <= hats[i].length <= 40 1 <= hats[i][j] <= 40 hats[i] contains a list of unique integers.

==

Problem Number: 1906 URL: https://leetcode.com/problems/find-the-kth-smallest-sum-of-a-matrix-with-sorted-rows Title: 1439. Find the Kth Smallest Sum of a Matrix With Sorted Rows Problem Description: You are given an m x n matrix mat that has its rows sorted in non-decreasing order and an integer k. You are allowed to choose exactly one element from each row to form an array. Return the kth smallest array sum among all possible arrays. Example 1: Input: mat = [[1,3,11],[2,4,6]], k = 5 Output: 7 Explanation: Choosing one element from each row, the first k smallest sum are: [1,2], [1,4], [3,2], [3,4], [1,6]. Where the 5th sum is 7.

Example 2: Input: mat = [[1,3,11],[2,4,6]], k = 9 Output: 17

Example 3: Input: mat = [[1,10,10],[1,4,5],[2,3,6]], k = 7 Output: 9 Explanation: Choosing one element from each row, the first k smallest sum are: [1,1,2], [1,1,3], [1,4,2], [1,4,3], [1,1,6], [1,5,2], [1,5,3]. Where the 7th sum is 9.

Constraints:

m == mat.length n == mat.length[i] 1 <= m, n <= 40 1 <= mat[i][j] <= 5000 1 <= k <= min(200, nm) mat[i] is a non-decreasing array.

==
Problem Number: 1907 URL: https://leetcode.com/problems/number-of-ways-of-cutting-a-pizza Title: 1444. Number of Ways of Cutting a Pizza Problem Description: Given a rectangular pizza represented as a rows x cols matrix containing the following characters: 'A' (an apple) and '.' (empty cell) and given the integer k. You have to cut the pizza into k pieces using k-1 cuts. For each cut you choose the direction: vertical or horizontal, then you choose a cut position at the cell boundary and cut the pizza into two pieces. If you cut the pizza vertically, give the left part of the pizza to a person. If you cut the pizza horizontally, give the upper part of the pizza to a person. Give the last piece of pizza to the last person. Return the number of ways of cutting the pizza such that each piece contains at least one apple. Since the answer can be a huge number, return this modulo 10^9 + 7. Example 1:

Input: pizza = ["A..","AAA","..."], k = 3 Output: 3 Explanation: The figure above shows the three ways to cut the pizza. Note that pieces must contain at least one apple.

Example 2: Input: pizza = ["A..","AA.","..."], k = 3 Output: 1

Example 3: Input: pizza = ["A..","A..","..."], k = 1 Output: 1

Constraints:

1 <= rows, cols <= 50 rows == pizza.length cols == pizza[i].length 1 <= k <= 10 pizza consists of characters 'A' and '.' only.

==
Problem Number: 1908 URL: https://leetcode.com/problems/form-largest-integer-with-digits-that-add-up-to-target Title: 1449. Form Largest Integer With Digits That Add up to Target Problem Description: Given an array of integers cost and an integer target, return the maximum integer you can paint under the following rules:

The cost of painting a digit (i + 1) is given by cost[i] (0-indexed). The total cost used must be equal to target. The integer does not have 0 digits.

Since the answer may be very large, return it as a string. If there is no way to paint any integer given the condition, return "0". Example 1: Input: cost = [4,3,2,5,6,7,2,5,5], target = 9 Output: "7772" Explanation: The cost to paint the digit '7' is 2, and the digit '2' is 3. Then cost("7772") = 2*3+ 3*1 = 9. You could also paint "977", but "7772" is the largest number. Digit cost 1 -> 4 2 -> 3 3 -> 2 4 -> 5 5 -> 6 6 -> 7 7 -> 2 8 -> 5 9 -> 5

Example 2: Input: cost = [7,6,5,5,5,6,8,7,8], target = 12 Output: "85" Explanation: The cost to paint the digit '8' is 7, and the digit '5' is 5. Then cost("85") = 7 + 5 = 12.

Example 3: Input: cost = [2,4,6,2,4,6,4,4,4], target = 5 Output: "0" Explanation: It is impossible to paint any integer with total cost equal to target.

Constraints:

cost.length == 9 1 <= cost[i], target <= 5000

==

Problem Number: 1909 URL: https://leetcode.com/problems/maximum-number-of-darts-inside-of-a-circular-dartboard Title: 1453. Maximum Number of Darts Inside of a Circular Dartboard Problem Description: Alice is throwing n darts on a very large wall. You are given an array darts where darts[i] = [xi, yi] is the position of the ith dart that Alice threw on the wall. Bob knows the positions of the n darts on the wall. He wants to place a dartboard of radius r on the wall so that the maximum number of darts that Alice throws lie on the dartboard. Given the integer r, return the maximum number of darts that can lie on the dartboard. Example 1:

Input: darts = [[-2,0],[2,0],[0,2],[0,-2]], r = 2 Output: 4 Explanation: Circle dartboard with center in (0,0) and radius = 2 contain all points.

Example 2:

Input: darts = [[-3,0],[3,0],[2,6],[5,4],[0,9],[7,8]], r = 5 Output: 5 Explanation: Circle dartboard with center in (0,4) and radius = 5 contain all points except the point (7,8).

Constraints:

1 <= darts.length <= 100 darts[i].length == 2 -104 <= xi, yi <= 104 All the darts are unique 1 <= r <= 5000

==

Problem Number: 1910 URL: https://leetcode.com/problems/max-dot-product-of-two-subsequences Title: 1458. Max Dot Product of Two Subsequences Problem Description: Given two arrays nums1 and nums2. Return the maximum dot product between non-empty subsequences of nums1 and nums2 with the same length. A subsequence of a array is a new array which is formed from the original array by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, [2,3,5] is a subsequence of [1,2,3,4,5] while [1,5,3] is not). Example 1: Input: nums1 = [2,1,-2,5], nums2 = [3,0,-6] Output: 18 Explanation: Take subsequence [2,-2] from nums1 and subsequence [3,-6] from nums2. Their dot product is (2*3 + (-2)*(-6)) = 18. Example 2: Input: nums1 = [3,-2], nums2 = [2,-6,7] Output: 21 Explanation: Take subsequence [3] from nums1 and subsequence [7] from nums2. Their dot product is (3*7) = 21. Example 3: Input: nums1 = [-1,-1], nums2 = [1,1] Output: -1 Explanation: Take subsequence [-1] from nums1 and subsequence [1] from nums2. Their dot product is -1. Constraints:

1 <= nums1.length, nums2.length <= 500 -1000 <= nums1[i], nums2[i] <= 1000

==
Problem Number: 1911 URL: https://leetcode.com/problems/cherry-pickup-ii Title: 1463. Cherry Pickup II Problem Description: You are given a rows x cols matrix grid representing a field of cherries where grid[i][j] represents the number of cherries that you can collect from the (i, j) cell. You have two robots that can collect cherries for you:

Robot #1 is located at the top-left corner (0, 0), and Robot #2 is located at the top-right corner (0, cols - 1).

Return the maximum number of cherries collection using both robots by following the rules below:

From a cell (i, j), robots can move to cell (i + 1, j - 1), (i + 1, j), or (i + 1, j + 1). When any robot passes through a cell, It picks up all cherries, and the cell becomes an empty cell. When both robots stay in the same cell, only one takes the cherries. Both robots cannot move outside of the grid at any moment. Both robots should reach the bottom row in grid.

Example 1:

Input: grid = [[3,1,1],[2,5,1],[1,5,5],[2,1,1]] Output: 24 Explanation: Path of robot #1 and #2 are described in color green and blue respectively. Cherries taken by Robot #1, (3 + 2 + 5 + 2) = 12. Cherries taken by Robot #2, (1 + 5 + 5 + 1) = 12. Total of cherries: 12 + 12 = 24.

Example 2:

Input: grid = [[1,0,0,0,0,0,1],[2,0,0,0,0,3,0],[2,0,9,0,0,0,0],[0,3,0,5,4,0,0],[1,0,2,3,0,0,6]] Output: 28 Explanation: Path of robot #1 and #2 are described in color green and blue respectively. Cherries taken by Robot #1, (1 + 9 + 5 + 2) = 17. Cherries taken by Robot #2, (1 + 3 + 4 + 3) = 11. Total of cherries: 17 + 11 = 28.

Constraints:

rows == grid.length cols == grid[i].length 2 <= rows, cols <= 70 0 <= grid[i][j] <= 100

==
Problem Number: 1912 URL: https://leetcode.com/problems/probability-of-a-two-boxes-having-the-same-number-of-distinct-balls Title: 1467. Probability of a Two Boxes Having The Same Number of Distinct Balls Problem Description: Given 2n balls of k distinct colors. You will be given an integer array balls of size k where balls[i] is the number of balls of color i. All the balls will be shuffled uniformly at random, then we will distribute the first n balls to the first box and the remaining n balls to the other box (Please read the explanation of the second example carefully). Please note that the two boxes are considered different. For example, if we have two balls of colors a and b, and two boxes [] and (), then the distribution [a] (b) is considered different than the distribution [b] (a) (Please read the explanation of the first example carefully). Return the probability that the two boxes have the same number of distinct balls. Answers within 10-5 of the actual value will be accepted as correct. Example 1: Input: balls = [1,1] Output: 1.00000 Explanation: Only 2 ways to divide the balls equally: - A ball of color 1 to box 1 and a ball of color 2 to box 2 - A ball of color 2 to box 1 and a ball of color 1 to box 2 In both ways, the number of distinct colors in each box is equal. The probability is 2/2 = 1

Example 2: Input: balls = [2,1,1] Output: 0.66667 Explanation: We have the set of balls [1, 1, 2, 3] This set of balls will be shuffled randomly and we may have one of the 12 distinct shuffles with equal probability (i.e. 1/12): [1,1 / 2,3], [1,1 / 3,2], [1,2 / 1,3], [1,2 / 3,1], [1,3 / 1,2], [1,3 / 2,1], [2,1 / 1,3], [2,1 / 3,1], [2,3 / 1,1], [3,1 / 1,2], [3,1 / 2,1], [3,2 / 1,1] After that, we add the first two balls to the first box and the second two balls to the second box. We can see that 8 of these 12 possible random distributions have the same number of distinct colors of balls in each box. Probability is 8/12 = 0.66667

Example 3: Input: balls = [1,2,1,2] Output: 0.60000 Explanation: The set of balls is [1, 2, 2, 3, 4, 4]. It is hard to display all the 180 possible random shuffles of this set but it is easy to check that 108 of them will have the same number of distinct colors in each box. Probability = 108 / 180 = 0.6

Constraints:

1 <= balls.length <= 8 1 <= balls[i] <= 6 sum(balls) is even.

==
Problem Number: 1913 URL: https://leetcode.com/problems/paint-house-iii Title: 1473. Paint House III Problem Description: There is a row of m houses in a small city, each house must be painted with one of the n colors (labeled from 1 to n), some houses that have been painted last summer should not be painted again. A neighborhood is a maximal group of continuous houses that are painted with the same color.

For example: houses = [1,2,2,3,3,2,1,1] contains 5 neighborhoods [{1}, {2,2}, {3,3}, {2}, {1,1}].

Given an array houses, an m x n matrix cost and an integer target where:

houses[i]: is the color of the house i, and 0 if the house is not painted yet. cost[i][j]: is the cost of paint the house i with the color j + 1.

Return the minimum cost of painting all the remaining houses in such a way that there are exactly target neighborhoods. If it is not possible, return -1. Example 1: Input: houses = [0,0,0,0,0], cost = [[1,10],[10,1],[10,1],[1,10],[5,1]], m = 5, n = 2, target = 3 Output: 9 Explanation: Paint houses of this way [1,2,2,1,1] This array contains target = 3 neighborhoods, [{1}, {2,2}, {1,1}]. Cost of paint all houses (1 + 1 + 1 + 1 + 5) = 9.

Example 2: Input: houses = [0,2,1,2,0], cost = [[1,10],[10,1],[10,1],[1,10],[5,1]], m = 5, n = 2, target = 3 Output: 11 Explanation: Some houses are already painted, Paint the houses of this way [2,2,1,2,2] This array contains target = 3 neighborhoods, [{2,2}, {1}, {2,2}]. Cost of paint the first and last house (10 + 1) = 11.

Example 3: Input: houses = [3,1,2,3], cost = [[1,1,1],[1,1,1],[1,1,1],[1,1,1]], m = 4, n = 3, target = 3 Output: -1 Explanation: Houses are already painted with a total of 4 neighborhoods [{3},{1},{2},{3}] different of target = 3.

Constraints:

m == houses.length == cost.length n == cost[i].length 1 <= m <= 100 1 <= n <= 20 1 <= target <= m 0 <= houses[i] <= n 1 <= cost[i][j] <= 104

==
Problem Number: 1914 URL: https://leetcode.com/problems/allocate-mailboxes Title: 1478. Allocate Mailboxes Problem Description: Given the array houses where houses[i] is the location of the ith house along a street and an integer k, allocate k mailboxes in the street. Return the minimum total distance between each house and its nearest mailbox. The test cases are generated so that the answer fits in a 32-bit integer. Example 1:

Input: houses = [1,4,8,10,20], k = 3 Output: 5 Explanation: Allocate mailboxes in position 3, 9 and 20. Minimum total distance from each houses to nearest mailboxes is |3-1| + |4-3| + |9-8| + |10-9| + |20-20| = 5

Example 2:

Input: houses = [2,3,5,12,18], k = 2 Output: 9 Explanation: Allocate mailboxes in position 3 and 14. Minimum total distance from each houses to nearest mailboxes is |2-3| + |3-3| + |5-3| + |12-14| + |18-14| = 9.

Constraints:

1 <= k <= houses.length <= 100 1 <= houses[i] <= 104 All the integers of houses are unique.

==
Problem Number: 1915 URL: https://leetcode.com/problems/kth-ancestor-of-a-tree-node Title: 1483. Kth Ancestor of a Tree Node Problem Description: You are given a tree with n nodes numbered from 0 to n - 1 in the form of

a parent array parent where parent[i] is the parent of ith node. The root of the tree is node 0. Find the kth ancestor of a given node. The kth ancestor of a tree node is the kth node in the path from that node to the root node. Implement the TreeAncestor class:

TreeAncestor(int n, int[] parent) Initializes the object with the number of nodes in the tree and the parent array. int getKthAncestor(int node, int k) return the kth ancestor of the given node node. If there is no such ancestor, return -1.

Example 1:

Input ["TreeAncestor", "getKthAncestor", "getKthAncestor", "getKthAncestor"] [[7, [-1, 0, 0, 1, 1, 2, 2]], [3, 1], [5, 2], [6, 3]] Output [null, 1, 0, -1]

Explanation TreeAncestor treeAncestor = new TreeAncestor(7, [-1, 0, 0, 1, 1, 2, 2]); treeAncestor.getKthAncestor(3, 1); // returns 1 which is the parent of 3 treeAncestor.getKthAncestor(5, 2); // returns 0 which is the grandparent of 5 treeAncestor.getKthAncestor(6, 3); // returns -1 because there is no such ancestor Constraints:

1 <= k <= n <= 5 * 104 parent.length == n parent[0] == -1 0 <= parent[i] < n for all 0 < i < n 0 <= node < n There will be at most 5 * 104 queries.

==
Problem Number: 1916 URL: https://leetcode.com/problems/find-critical-and-pseudo-critical-edges-in-minimum-spanning-tree Title: 1489. Find Critical and Pseudo-Critical Edges in Minimum Spanning Tree Problem Description: Given a weighted undirected connected graph with n vertices numbered from 0 to n - 1, and an array edges where edges[i] = [ai, bi, weighti] represents a bidirectional and weighted edge between nodes ai and bi. A minimum spanning tree (MST) is a subset of the graph's edges that connects all vertices without cycles and with the minimum possible total edge weight. Find all the critical and pseudo-critical edges in the given graph's minimum spanning tree (MST). An MST edge whose deletion from the graph would cause the MST weight to increase is called a critical edge. On the other hand, a pseudo-critical edge is that which can appear in some MSTs but not all. Note that you can return the indices of the edges in any order. Example 1:

Input: n = 5, edges = [[0,1,1],[1,2,1],[2,3,2],[0,3,2],[0,4,3],[3,4,3],[1,4,6]] Output: [[0,1],[2,3,4,5]] Explanation: The figure above describes the graph. The following figure shows all the possible MSTs:

Notice that the two edges 0 and 1 appear in all MSTs, therefore they are critical edges, so we return them in the first list of the output. The edges 2, 3, 4, and 5 are only part of some MSTs, therefore they are considered pseudo-critical edges. We add them to the second list of the output.

Example 2:

Input: n = 4, edges = [[0,1,1],[1,2,1],[2,3,1],[0,3,1]] Output: [[],[0,1,2,3]] Explanation: We can observe that since all 4 edges have equal weight, choosing any

3 edges from the given 4 will yield an MST. Therefore all 4 edges are pseudo-critical.

Constraints:

2 <= n <= 100 1 <= edges.length <= min(200, n * (n - 1) / 2) edges[i].length == 3 0 <= ai < bi < n 1 <= weighti <= 1000 All pairs (ai, bi) are distinct.

==
Problem Number: 1917 URL: https://leetcode.com/problems/parallel-courses-ii Title: 1494. Parallel Courses II Problem Description: You are given an integer n, which indicates that there are n courses labeled from 1 to n. You are also given an array relations where relations[i] = [prevCoursei, nextCoursei], representing a prerequisite relationship between course prevCoursei and course nextCoursei: course prevCoursei has to be taken before course nextCoursei. Also, you are given the integer k. In one semester, you can take at most k courses as long as you have taken all the prerequisites in the previous semesters for the courses you are taking. Return the minimum number of semesters needed to take all courses. The testcases will be generated such that it is possible to take every course. Example 1:

Input: n = 4, relations = [[2,1],[3,1],[1,4]], k = 2 Output: 3 Explanation: The figure above represents the given graph. In the first semester, you can take courses 2 and 3. In the second semester, you can take course 1. In the third semester, you can take course 4.

Example 2:

Input: n = 5, relations = [[2,1],[3,1],[4,1],[1,5]], k = 2 Output: 4 Explanation: The figure above represents the given graph. In the first semester, you can only take courses 2 and 3 since you cannot take more than two per semester. In the second semester, you can take course 4. In the third semester, you can take course 1. In the fourth semester, you can take course 5.

Constraints:

1 <= n <= 15 1 <= k <= n 0 <= relations.length <= n * (n-1) / 2 relations[i].length == 2 1 <= prevCoursei, nextCoursei <= n prevCoursei != nextCoursei All the pairs [prevCoursei, nextCoursei] are unique. The given graph is a directed acyclic graph.

==
Problem Number: 1918 URL: https://leetcode.com/problems/max-value-of-equation Title: 1499. Max Value of Equation Problem Description: You are given an array points containing the coordinates of points on a 2D plane, sorted by the x-values, where points[i] = [xi, yi] such that xi < xj for all 1 <= i < j <= points.length. You are also given an integer k. Return the maximum value of the equation yi + yj + |xi - xj| where |xi - xj| <= k and 1 <= i < j <= points.length. It is guaranteed that there exists at least one pair of points that satisfy the constraint |xi - xj| <= k. Example 1: Input: points

= [[1,3],[2,0],[5,10],[6,-10]], k = 1 Output: 4 Explanation: The first two points satisfy the condition |xi - xj| <= 1 and if we calculate the equation we get 3 + 0 + |1 - 2| = 4. Third and fourth points also satisfy the condition and give a value of 10 + -10 + |5 - 6| = 1. No other pairs satisfy the condition, so we return the max of 4 and 1.

Example 2: Input: points = [[0,0],[3,0],[9,2]], k = 3 Output: 3 Explanation: Only the first two points have an absolute difference of 3 or less in the x-values, and give the value of 0 + 0 + |0 - 3| = 3.

Constraints:

2 <= points.length <= 105 points[i].length == 2 -108 <= xi, yi <= 108 0 <= k <= 2 * 108 xi < xj for all 1 <= i < j <= points.length xi form a strictly increasing sequence.

==
Problem Number: 1919 URL: https://leetcode.com/problems/minimum-possible-integer-after-at-most-k-adjacent-swaps-on-digits Title: 1505. Minimum Possible Integer After at Most K Adjacent Swaps On Digits Problem Description: You are given a string num representing the digits of a very large integer and an integer k. You are allowed to swap any two adjacent digits of the integer at most k times. Return the minimum integer you can obtain also as a string. Example 1:

Input: num = "4321", k = 4 Output: "1342" Explanation: The steps to obtain the minimum integer from 4321 with 4 adjacent swaps are shown.

Example 2: Input: num = "100", k = 1 Output: "010" Explanation: It's ok for the output to have leading zeros, but the input is guaranteed not to have any leading zeros.

Example 3: Input: num = "36789", k = 1000 Output: "36789" Explanation: We can keep the number without any swaps.

Constraints:

1 <= num.length <= 3 * 104 num consists of only digits and does not contain leading zeros. 1 <= k <= 109

==
Problem Number: 1920 URL: https://leetcode.com/problems/stone-game-iv Title: 1510. Stone Game IV Problem Description: Alice and Bob take turns playing a game, with Alice starting first. Initially, there are n stones in a pile. On each player's turn, that player makes a move consisting of removing any non-zero square number of stones in the pile. Also, if a player cannot make a move, he/she loses the game. Given a positive integer n, return true if and only if Alice wins the game otherwise return false, assuming both players play optimally. Example 1: Input: n = 1 Output: true Explanation: Alice can remove 1 stone winning the game because Bob doesn't have any moves.

Example 2: Input: n = 2 Output: false Explanation: Alice can only remove 1 stone, after that Bob removes the last one winning the game (2 -> 1 -> 0).

Example 3: Input: n = 4 Output: true Explanation: n is already a perfect square, Alice can win with one move, removing 4 stones (4 -> 0).

Constraints:

1 <= n <= 105

==

Problem Number: 1921 URL: https://leetcode.com/problems/best-position-for-a-service-centre Title: 1515. Best Position for a Service Centre Problem Description: A delivery company wants to build a new service center in a new city. The company knows the positions of all the customers in this city on a 2D-Map and wants to build the new center in a position such that the sum of the euclidean distances to all customers is minimum. Given an array positions where positions[i] = [xi, yi] is the position of the ith customer on the map, return the minimum sum of the euclidean distances to all customers. In other words, you need to choose the position of the service center [xcentre, ycentre] such that the following formula is minimized:

Answers within 10-5 of the actual value will be accepted. Example 1:

Input: positions = [[0,1],[1,0],[1,2],[2,1]] Output: 4.00000 Explanation: As shown, you can see that choosing [xcentre, ycentre] = [1, 1] will make the distance to each customer = 1, the sum of all distances is 4 which is the minimum possible we can achieve.

Example 2:

Input: positions = [[1,1],[3,3]] Output: 2.82843 Explanation: The minimum possible sum of distances = sqrt(2) + sqrt(2) = 2.82843

Constraints:

1 <= positions.length <= 50 positions[i].length == 2 0 <= xi, yi <= 100

==

Problem Number: 1922 URL: https://leetcode.com/problems/maximum-number-of-non-overlapping-substrings Title: 1520. Maximum Number of Non-Overlapping Substrings Problem Description: Given a string s of lowercase letters, you need to find the maximum number of non-empty substrings of s that meet the following conditions:

The substrings do not overlap, that is for any two substrings s[i..j] and s[x..y], either $j < x$ or $i > y$ is true. A substring that contains a certain character c must also contain all occurrences of c.

Find the maximum number of substrings that meet the above conditions. If there are multiple solutions with the same number of substrings, return the one with minimum total length. It can be shown that there exists a unique solution of minimum total length. Notice that you can return the substrings in any order. Example 1: Input: s = "adefaddaccc" Output: ["e","f","ccc"] Explanation: The following are all the possible substrings that meet the conditions: ["adefaddaccc" "adefadda", "ef", "e", "f", "ccc",] If we choose the first string, we cannot choose anything else and we'd get only 1. If we choose "adefadda", we are left with "ccc" which is the only one that doesn't overlap, thus obtaining 2 substrings. Notice also, that it's not optimal to choose "ef" since it can be split into two. Therefore, the optimal way is to choose ["e","f","ccc"] which gives us 3 substrings. No other solution of the same number of substrings exist.

Example 2: Input: s = "abbaccd" Output: ["d","bb","cc"] Explanation: Notice that while the set of substrings ["d","abba","cc"] also has length 3, it's considered incorrect since it has larger total length.

Constraints:

1 <= s.length <= 105 s contains only lowercase English letters.

==
Problem Number: 1923 URL: https://leetcode.com/problems/find-a-value-of-a-mysterious-function-closest-to-target Title: 1521. Find a Value of a Mysterious Function Closest to Target Problem Description:

Winston was given the above mysterious function func. He has an integer array arr and an integer target and he wants to find the values l and r that make the value |func(arr, l, r) - target| minimum possible. Return the minimum possible value of |func(arr, l, r) - target|. Notice that func should be called with the values l and r where 0 <= l, r < arr.length. Example 1: Input: arr = [9,12,3,7,15], target = 5 Output: 2 Explanation: Calling func with all the pairs of [l,r] = [[0,0],[1,1],[2,2],[3,3],[4,4],[0,1],[1,2],[2,3],[3,4],[0,2],[1,3],[2,4],[0,3],[1,4],[0,4]], Winston got the following results [9,12,3,7,15,8,0,3,7,0,0,3,0,0,0]. The value closest to 5 is 7 and 3, thus the minimum difference is 2.

Example 2: Input: arr = [1000000,1000000,1000000], target = 1 Output: 999999 Explanation: Winston called the func with all possible values of [l,r] and he always got 1000000, thus the min difference is 999999.

Example 3: Input: arr = [1,2,4,8,16], target = 0 Output: 0

Constraints:

1 <= arr.length <= 105 1 <= arr[i] <= 106 0 <= target <= 107

==
Problem Number: 1924 URL: https://leetcode.com/problems/minimum-number-of-increments-on-subarrays-to-form-a-target-array Title: 1526. Minimum Number of Increments on Subarrays to Form a Target Array Problem Description: You are given an integer array target. You have an integer array initial of the same size as target with all elements initially zeros. In one operation you can choose any subarray from initial and increment each value by one. Return the minimum number of operations to form a target array from

initial. The test cases are generated so that the answer fits in a 32-bit integer. Example 1: Input: target = [1,2,3,2,1] Output: 3 Explanation: We need at least 3 operations to form the target array from the initial array. [0,0,0,0,0] increment 1 from index 0 to 4 (inclusive). [1,1,1,1,1] increment 1 from index 1 to 3 (inclusive). [1,2,2,2,1] increment 1 at index 2. [1,2,3,2,1] target array is formed.

Example 2: Input: target = [3,1,1,2] Output: 4 Explanation: [0,0,0,0] -> [1,1,1,1] -> [1,1,1,2] -> [2,1,1,2] -> [3,1,1,2]

Example 3: Input: target = [3,1,5,4,2] Output: 7 Explanation: [0,0,0,0,0] -> [1,1,1,1,1] -> [2,1,1,1,1] -> [3,1,1,1,1] -> [3,1,2,2,2] -> [3,1,3,3,2] -> [3,1,4,4,2] -> [3,1,5,4,2].

Constraints:

1 <= target.length <= 105 1 <= target[i] <= 105

==

Problem Number: 1925 URL: https://leetcode.com/problems/string-compression-ii Title: 1531. String Compression II Problem Description: Run-length encoding is a string compression method that works by replacing consecutive identical characters (repeated 2 or more times) with the concatenation of the character and the number marking the count of the characters (length of the run). For example, to compress the string "aabccc" we replace "aa" by "a2" and replace "ccc" by "c3". Thus the compressed string becomes "a2bc3". Notice that in this problem, we are not adding '1' after single characters. Given a string s and an integer k. You need to delete at most k characters from s such that the run-length encoded version of s has minimum length. Find the minimum length of the run-length encoded version of s after deleting at most k characters. Example 1: Input: s = "aaabcccd", k = 2 Output: 4 Explanation: Compressing s without deleting anything will give us "a3bc3d" of length 6. Deleting any of the characters 'a' or 'c' would at most decrease the length of the compressed string to 5, for instance delete 2 'a' then we will have s = "abcccd" which compressed is abc3d. Therefore, the optimal way is to delete 'b' and 'd', then the compressed version of s will be "a3c3" of length 4. Example 2: Input: s = "aabbaa", k = 2 Output: 2 Explanation: If we delete both 'b' characters, the resulting compressed string would be "a4" of length 2.

Example 3: Input: s = "aaaaaaaaaaa", k = 0 Output: 3 Explanation: Since k is zero, we cannot delete anything. The compressed string is "a11" of length 3.

Constraints:

1 <= s.length <= 100 0 <= k <= s.length s contains only lowercase English letters.

==

Problem Number: 1926 URL: https://leetcode.com/problems/get-the-maximum-score Title: 1537. Get the Maximum Score Problem Description: You are given two sorted arrays of distinct integers nums1 and nums2. A valid path is defined as follows:

Choose array nums1 or nums2 to traverse (from index-0). Traverse the current array from left to right. If you are reading any value that is present in nums1 and nums2 you are allowed to change your path to the other array. (Only one repeated value is considered in the valid path).

The score is defined as the sum of uniques values in a valid path. Return the maximum score you can obtain of all possible valid paths. Since the answer may be too large, return it modulo 109 + 7. Example 1:

Input: nums1 = [2,4,5,8,10], nums2 = [4,6,8,9] Output: 30 Explanation: Valid paths: [2,4,5,8,10], [2,4,5,8,9], [2,4,6,8,9], [2,4,6,8,10], (starting from nums1) [4,6,8,9], [4,5,8,10], [4,5,8,9], [4,6,8,10] (starting from nums2) The maximum is obtained with the path in green [2,4,6,8,10].

Example 2: Input: nums1 = [1,3,5,7,9], nums2 = [3,5,100] Output: 109 Explanation: Maximum sum is obtained with the path [1,3,5,100].

Example 3: Input: nums1 = [1,2,3,4,5], nums2 = [6,7,8,9,10] Output: 40 Explanation: There are no common elements between nums1 and nums2. Maximum sum is obtained with the path [6,7,8,9,10].

Constraints:

1 <= nums1.length, nums2.length <= 105 1 <= nums1[i], nums2[i] <= 107 nums1 and nums2 are strictly increasing.

==
Problem Number: 1927 URL: https://leetcode.com/problems/find-longest-awesome-substring Title: 1542. Find Longest Awesome Substring Problem Description: You are given a string s. An awesome substring is a non-empty substring of s such that we can make any number of swaps in order to make it a palindrome. Return the length of the maximum length awesome substring of s. Example 1: Input: s = ”3242415” Output: 5 Explanation: ”24241” is the longest awesome substring, we can form the palindrome ”24142” with some swaps.

Example 2: Input: s = ”12345678” Output: 1

Example 3: Input: s = ”213123” Output: 6 Explanation: ”213123” is the longest awesome substring, we can form the palindrome ”231132” with some swaps.

Constraints:

1 <= s.length <= 105 s consists only of digits.

==
Problem Number: 1928 URL: https://leetcode.com/problems/minimum-cost-to-cut-a-stick Title: 1547. Minimum Cost to Cut a Stick Problem Description:

Given a wooden stick of length n units. The stick is labelled from 0 to n. For example, a stick of length 6 is labelled as follows:

Given an integer array cuts where cuts[i] denotes a position you should perform a cut at. You should perform the cuts in order, you can change the order of the cuts as you wish. The cost of one cut is the length of the stick to be cut, the total cost is the sum of costs of all cuts. When you cut a stick, it will be split into two smaller sticks (i.e. the sum of their lengths is the length of the stick before the cut). Please refer to the first example for a better explanation. Return the minimum total cost of the cuts. Example 1:

Input: n = 7, cuts = [1,3,4,5] Output: 16 Explanation: Using cuts order = [1, 3, 4, 5] as in the input leads to the following scenario:

The first cut is done to a rod of length 7 so the cost is 7. The second cut is done to a rod of length 6 (i.e. the second part of the first cut), the third is done to a rod of length 4 and the last cut is to a rod of length 3. The total cost is 7 + 6 + 4 + 3 = 20. Rearranging the cuts to be [3, 5, 1, 4] for example will lead to a scenario with total cost = 16 (as shown in the example photo 7 + 4 + 3 + 2 = 16). Example 2: Input: n = 9, cuts = [5,6,1,4,2] Output: 22 Explanation: If you try the given cuts ordering the cost will be 25. There are much ordering with total cost <= 25, for example, the order [4, 6, 5, 2, 1] has total cost = 22 which is the minimum possible.

Constraints:

2 <= n <= 106 1 <= cuts.length <= min(n - 1, 100) 1 <= cuts[i] <= n - 1 All the integers in cuts array are distinct.

==
Problem Number: 1929 URL: https://leetcode.com/problems/minimum-number-of-days-to-eat-n-oranges Title: 1553. Minimum Number of Days to Eat N Oranges Problem Description: There are n oranges in the kitchen and you decided to eat some of these oranges every day as follows:

Eat one orange. If the number of remaining oranges n is divisible by 2 then you can eat n / 2 oranges. If the number of remaining oranges n is divisible by 3 then you can eat 2 * (n / 3) oranges.

You can only choose one of the actions per day. Given the integer n, return the minimum number of days to eat n oranges. Example 1: Input: n = 10 Output: 4 Explanation: You have 10 oranges. Day 1: Eat 1 orange, 10 - 1 = 9. Day 2: Eat 6 oranges, 9 - 2*(9/3) = 9 - 6 = 3. (Since 9 is divisible by 3) Day 3: Eat 2 oranges, 3 - 2*(3/3) = 3 - 2 = 1. Day 4: Eat the last orange 1 - 1 = 0. You need at least 4 days to eat the 10 oranges.

Example 2: Input: n = 6 Output: 3 Explanation: You have 6 oranges. Day 1: Eat 3 oranges, 6 - 6/2 = 6 - 3 = 3. (Since 6 is divisible by 2). Day 2: Eat 2 oranges, 3 - 2*(3/3) = 3 - 2 = 1. (Since 3 is divisible by 3) Day 3: Eat the last orange 1 - 1 = 0. You need at least 3 days to eat the 6 oranges.

Constraints:

1 <= n <= 2 * 109

==
Problem Number: 1930 URL: https://leetcode.com/problems/stone-game-v Title: 1563. Stone Game V Problem Description: There are several stones arranged in a row, and each stone has an associated value which is an integer given in the array stoneValue. In each round of the game, Alice divides the row into two non-empty rows (i.e. left row and right row), then Bob calculates the value of each row which is the sum of the values of all the stones in this row. Bob throws away the row which has the maximum value, and Alice's score increases by the value of the remaining row. If the value of the two rows are equal, Bob lets Alice decide which row will be thrown away. The next round starts with the remaining row. The game ends when there is only one stone remaining. Alice's is initially zero. Return the maximum score that Alice can obtain. Example 1: Input: stoneValue = [6,2,3,4,5,5] Output: 18 Explanation: In the first round, Alice divides the row to [6,2,3], [4,5,5]. The left row has the value 11 and the right row has value 14. Bob throws away the right row and Alice's score is now 11. In the second round Alice divides the row to [6], [2,3]. This time Bob throws away the left row and Alice's score becomes 16 (11 + 5). The last round Alice has only one choice to divide the row which is [2], [3]. Bob throws away the right row and Alice's score is now 18 (16 + 2). The game ends because only one stone is remaining in the row.

Example 2: Input: stoneValue = [7,7,7,7,7,7,7] Output: 28

Example 3: Input: stoneValue = [4] Output: 0

Constraints:

1 <= stoneValue.length <= 500 1 <= stoneValue[i] <= 106

==
Problem Number: 1931 URL: https://leetcode.com/problems/minimum-number-of-days-to-disconnect-island Title: 1568. Minimum Number of Days to Disconnect Island Problem Description: You are given an m x n binary grid grid where 1 represents land and 0 represents water. An island is a maximal 4-directionally (horizontal or vertical) connected group of 1's. The grid is said to be connected if we have exactly one island, otherwise is said disconnected. In one day, we are allowed to change any single land cell (1) into a water cell (0). Return the minimum number of days to disconnect the grid. Example 1:

Input: grid = [[0,1,1,0],[0,1,1,0],[0,0,0,0]]

Output: 2 Explanation: We need at least 2 days to get a disconnected grid. Change land grid[1][1] and grid[0][2] to water and get 2 disconnected island.

Example 2:

Input: grid = [[1,1]] Output: 2 Explanation: Grid of full water is also disconnected ([[1,1]] -> [[0,0]]), 0 islands.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 30 grid[i][j] is either 0 or 1.

==

Problem Number: 1932 URL: https://leetcode.com/problems/number-of-ways-to-reorder-array-to-get-same-bst Title: 1569. Number of Ways to Reorder Array to Get Same BST Problem Description: Given an array nums that represents a permutation of integers from 1 to n. We are going to construct a binary search tree (BST) by inserting the elements of nums in order into an initially empty BST. Find the number of different ways to reorder nums so that the constructed BST is identical to that formed from the original array nums.

For example, given nums = [2,1,3], we will have 2 as the root, 1 as a left child, and 3 as a right child. The array [2,3,1] also yields the same BST but [3,2,1] yields a different BST.

Return the number of ways to reorder nums such that the BST formed is identical to the original BST formed from nums. Since the answer may be very large, return it modulo 109 + 7. Example 1:

Input: nums = [2,1,3] Output: 1 Explanation: We can reorder nums to be [2,3,1] which will yield the same BST. There are no other ways to reorder nums which will yield the same BST.

Example 2:

Input: nums = [3,4,5,1,2] Output: 5 Explanation: The following 5 arrays will yield the same BST: [3,1,2,4,5] [3,1,4,2,5] [3,1,4,5,2] [3,4,1,2,5] [3,4,1,5,2]

Example 3:

Input: nums = [1,2,3] Output: 0 Explanation: There are no other orderings of nums that will yield the same BST.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= nums.length All integers in nums are distinct.

==

Problem Number: 1933 URL: https://leetcode.com/problems/count-all-possible-routes Title: 1575. Count All Possible Routes Problem Description: You are given an array of distinct positive integers locations where locations[i] represents the position of city i. You are also given integers start, finish and fuel representing the starting city, ending city, and the initial amount of fuel you have, respectively. At each step, if you are at city i, you can pick any city j such that j != i and 0 <= j < locations.length and move to city j. Moving from city i to city j reduces the amount of fuel you have by |locations[i] - locations[j]|. Please notice that |x| denotes the absolute value of x. Notice that fuel cannot become negative at any point in time, and that you are allowed to visit any city more than once (including start and finish). Return the count of all possible routes from start to finish. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: locations = [2,3,6,8,4], start = 1, finish = 3, fuel = 5 Output: 4 Explanation: The following are all possible routes, each uses 5 units of fuel: 1 -> 3 1 -> 2 -> 3 1 -> 4 -> 3 1 -> 4 -> 2 -> 3

Example 2: Input: locations = [4,3,1], start = 1, finish = 0, fuel = 6 Output: 5 Explanation: The following are all possible routes: 1 -> 0, used fuel = 1 1 -> 2 -> 0, used fuel = 5 1 -> 2 -> 1 -> 0, used fuel = 5 1 -> 0 -> 1 -> 0, used fuel = 3 1 -> 0 -> 1 -> 0 -> 1 -> 0, used fuel = 5

Example 3: Input: locations = [5,2,1], start = 0, finish = 2, fuel = 3 Output: 0 Explanation: It is impossible to get from 0 to 2 using only 3 units of fuel since the shortest route needs 4 units of fuel.

Constraints:

2 <= locations.length <= 100 1 <= locations[i] <= 109 All integers in locations are distinct. 0 <= start, finish < locations.length 1 <= fuel <= 200

==
Problem Number: 1934 URL: https://leetcode.com/problems/remove-max-number-of-edges-to-keep-graph-fully-traversable Title: 1579. Remove Max Number of Edges to Keep Graph Fully Traversable Problem Description: Alice and Bob have an undirected graph of n nodes and three types of edges:

Type 1: Can be traversed by Alice only. Type 2: Can be traversed by Bob only. Type 3: Can be traversed by both Alice and Bob.

Given an array edges where edges[i] = [typei, ui, vi] represents a bidirectional edge of type typei between nodes ui and vi, find the maximum number of edges you can remove so that after removing the edges, the graph can still be fully traversed by both Alice and Bob. The graph is fully traversed by Alice and Bob if starting from any node, they can reach all other nodes. Return the maximum number of edges you can remove, or return -1 if Alice and Bob cannot fully traverse the graph. Example 1:

Input: n = 4, edges = [[3,1,2],[3,2,3],[1,1,3],[1,2,4],[1,1,2],[2,3,4]] Output: 2 Explanation: If we remove the 2 edges [1,1,2] and [1,1,3]. The graph will still be fully traversable by Alice and Bob. Removing any additional edge will not make it so. So the maximum number of edges we can remove is 2.

Example 2:

Input: n = 4, edges = [[3,1,2],[3,2,3],[1,1,4],[2,1,4]] Output: 0 Explanation: Notice that removing any edge will not make the graph fully traversable by Alice and Bob.

Example 3:

Input: n = 4, edges = [[3,2,3],[1,1,2],[2,3,4]] Output: -1 Explanation: In the current graph, Alice cannot reach node 4 from the other nodes. Likewise, Bob cannot reach 1. Therefore it's impossible to make the graph fully traversable. Constraints:

1 <= n <= 105 1 <= edges.length <= min(105, 3 * n * (n - 1) / 2) edges[i].length == 3 1 <= typei <= 3 1 <= ui < vi <= n All tuples (typei, ui, vi) are distinct.

==

Problem Number: 1935 URL: https://leetcode.com/problems/check-if-string-is-transformable-with-substring-sort-operations Title: 1585. Check If String Is Transformable With Substring Sort Operations Problem Description: Given two strings s and t, transform string s into string t using the following operation any number of times:

Choose a non-empty substring in s and sort it in place so the characters are in ascending order.

For example, applying the operation on the underlined substring in "14234" results in "12344".

Return true if it is possible to transform s into t. Otherwise, return false. A substring is a contiguous sequence of characters within a string.   Example 1: Input: s = "84532", t = "34852" Output: true Explanation: You can transform s into t using the following sort operations: "84532" (from index 2 to 3) -> "84352" "84352" (from index 0 to 2) -> "34852"

Example 2: Input: s = "34521", t = "23415" Output: true Explanation: You can transform s into t using the following sort operations: "34521" -> "23451" "23451" -> "23415"

Example 3: Input: s = "12345", t = "12435" Output: false

Constraints:

s.length == t.length 1 <= s.length <= 105 s and t consist of only digits.

==

Problem Number: 1936 URL: https://leetcode.com/problems/strange-printer-ii Title: 1591. Strange Printer II Problem Description: There is a strange printer with the following two special requirements:

On each turn, the printer will print a solid rectangular pattern of a single color on the grid. This will cover up the existing colors in the rectangle. Once the printer has used a color for the above operation, the same color cannot be used again.

You are given a m x n matrix targetGrid, where targetGrid[row][col] is the color in the position (row, col) of the grid. Return true if it is possible to print the matrix targetGrid, otherwise, return false.   Example 1:

Input: targetGrid = [[1,1,1,1],[1,2,2,1],[1,2,2,1],[1,1,1,1]] Output: true

Example 2:

Input: targetGrid = [[1,1,1,1],[1,1,3,3],[1,1,3,4],[5,5,1,4]] Output: true

Example 3: Input: targetGrid = [[1,2,1],[2,1,2],[1,2,1]] Output: false Explanation: It is impossible to form targetGrid because it is not allowed to print the same color in different turns.

Constraints:

m == targetGrid.length n == targetGrid[i].length 1 <= m, n <= 60 1 <= targetGrid[row][col] <= 60

==
Problem Number: 1937 URL: https://leetcode.com/problems/minimum-cost-to-connect-two-groups-of-points Title: 1595. Minimum Cost to Connect Two Groups of Points Problem Description: You are given two groups of points where the first group has size1 points, the second group has size2 points, and size1 >= size2. The cost of the connection between any two points are given in an size1 x size2 matrix where cost[i][j] is the cost of connecting point i of the first group and point j of the second group. The groups are connected if each point in both groups is connected to one or more points in the opposite group. In other words, each point in the first group must be connected to at least one point in the second group, and each point in the second group must be connected to at least one point in the first group. Return the minimum cost it takes to connect the two groups. Example 1:

Input: cost = [[15, 96], [36, 2]] Output: 17 Explanation: The optimal way of connecting the groups is: 1--A 2--B This results in a total cost of 17.

Example 2:

Input: cost = [[1, 3, 5], [4, 1, 1], [1, 5, 3]] Output: 4 Explanation: The optimal way of connecting the groups is: 1--A 2--B 2--C 3--A This results in a total cost of 4. Note that there are multiple points connected to point 2 in the first group and point A in the second group. This does not matter as there is no limit to the number of points that can be connected. We only care about the minimum total cost.

Example 3: Input: cost = [[2, 5, 1], [3, 4, 7], [8, 1, 2], [6, 2, 4], [3, 8, 8]] Output: 10

Constraints:

size1 == cost.length size2 == cost[i].length 1 <= size1, size2 <= 12 size1 >= size2 0 <= cost[i][j] <= 100

==
Problem Number: 1938 URL: https://leetcode.com/problems/maximum-number-of-achievable-transfer-requests Title: 1601. Maximum Number of

Achievable Transfer Requests Problem Description: We have n buildings numbered from 0 to n - 1. Each building has a number of employees. It's transfer season, and some employees want to change the building they reside in. You are given an array requests where requests[i] = [fromi, toi] represents an employee's request to transfer from building fromi to building toi. All buildings are full, so a list of requests is achievable only if for each building, the net change in employee transfers is zero. This means the number of employees leaving is equal to the number of employees moving in. For example if n = 3 and two employees are leaving building 0, one is leaving building 1, and one is leaving building 2, there should be two employees moving to building 0, one employee moving to building 1, and one employee moving to building 2. Return the maximum number of achievable requests. Example 1:

Input: n = 5, requests = [[0,1],[1,0],[0,1],[1,2],[2,0],[3,4]] Output: 5 Explantion: Let's see the requests: From building 0 we have employees x and y and both want to move to building 1. From building 1 we have employees a and b and they want to move to buildings 2 and 0 respectively. From building 2 we have employee z and they want to move to building 0. From building 3 we have employee c and they want to move to building 4. From building 4 we don't have any requests. We can achieve the requests of users x and b by swapping their places. We can achieve the requests of users y, a and z by swapping the places in the 3 buildings.

Example 2:

Input: n = 3, requests = [[0,0],[1,2],[2,1]] Output: 3 Explantion: Let's see the requests: From building 0 we have employee x and they want to stay in the same building 0. From building 1 we have employee y and they want to move to building 2. From building 2 we have employee z and they want to move to building 1. We can achieve all the requests. Example 3: Input: n = 4, requests = [[0,3],[3,1],[1,2],[2,0]] Output: 4

Constraints:

1 <= n <= 20 1 <= requests.length <= 16 requests[i].length == 2 0 <= fromi, toi < n

==
Problem Number: 1939 URL: https://leetcode.com/problems/find-servers-that-handled-most-number-of-requests Title: 1606. Find Servers That Handled Most Number of Requests Problem Description: You have k servers numbered from 0 to k-1 that are being used to handle multiple requests simultaneously. Each server has infinite computational capacity but cannot handle more than one request at a time. The requests are assigned to servers according to a specific algorithm:

The ith (0-indexed) request arrives. If all servers are busy, the request is dropped (not handled at all). If the (i % k)th server is available, assign the request to that server. Otherwise, assign the request to the next available server (wrapping around the list of servers and starting from 0 if necessary). For example, if the ith server is busy, try to assign the request to the (i+1)th server, then the (i+2)th server, and so on.

You are given a strictly increasing array arrival of positive integers, where arrival[i] represents the arrival time of the ith request, and another array load, where load[i] represents the load of the ith request (the time it takes to complete). Your goal is to find the busiest server(s). A server is considered busiest if it handled the most number of requests successfully among all the servers. Return a list containing the IDs (0-indexed) of the busiest server(s). You may return the IDs in any order. Example 1:

Input: k = 3, arrival = [1,2,3,4,5], load = [5,2,3,3,3] Output: [1] Explanation: All of the servers start out available. The first 3 requests are handled by the first 3 servers in order. Request 3 comes in. Server 0 is busy, so it's assigned to the next available server, which is 1. Request 4 comes in. It cannot be handled since all servers are busy, so it is dropped. Servers 0 and 2 handled one request each, while server 1 handled two requests. Hence server 1 is the busiest server.

Example 2: Input: k = 3, arrival = [1,2,3,4], load = [1,2,1,2] Output: [0] Explanation: The first 3 requests are handled by first 3 servers. Request 3 comes in. It is handled by server 0 since the server is available. Server 0 handled two requests, while servers 1 and 2 handled one request each. Hence server 0 is the busiest server.

Example 3: Input: k = 3, arrival = [1,2,3], load = [10,12,11] Output: [0,1,2] Explanation: Each server handles a single request, so they are all considered the busiest.

Constraints:

1 <= k <= 105 1 <= arrival.length, load.length <= 105 arrival.length == load.length 1 <= arrival[i], load[i] <= 109 arrival is strictly increasing.

== Problem Number: 1940 URL: https://leetcode.com/problems/maximum-number-of-visible-points Title: 1610. Maximum Number of Visible Points Problem Description: You are given an array points, an integer angle, and your location, where location = [posx, posy] and points[i] = [xi, yi] both denote integral coordinates on the X-Y plane. Initially, you are facing directly east from your position. You cannot move from your position, but you can rotate. In other words, posx and posy cannot be changed. Your field of view in degrees is represented by angle, determining how wide you can see from any given view direction. Let d be the amount in degrees that you rotate counterclockwise. Then, your field of view is the inclusive range of angles [d - angle/2, d + angle/2].

Your browser does not support the video tag or this video format.

You can see some set of points if, for each point, the angle formed by the point, your position, and the immediate east direction from your position is in your field of view. There can be multiple points at one coordinate. There may be points at your location, and you can always see these points regardless of your rotation. Points do not obstruct your vision to other points. Return the maximum number of points you can see. Example 1:

Input: points = [[2,1],[2,2],[3,3]], angle = 90, location = [1,1] Output: 3 Explanation: The shaded region represents your field of view. All points can be made visible in your field of view, including [3,3] even though [2,2] is in front and in the same line of sight.

Example 2: Input: points = [[2,1],[2,2],[3,4],[1,1]], angle = 90, location = [1,1] Output: 4 Explanation: All points can be made visible in your field of view, including the one at your location.

Example 3:

Input: points = [[1,0],[2,1]], angle = 13, location = [1,1] Output: 1 Explanation: You can only see one of the two points, as shown above.

Constraints:

1 <= points.length <= 105 points[i].length == 2 location.length == 2 0 <= angle < 360 0 <= posx, posy, xi, yi <= 100

==
Problem Number: 1941 URL: https://leetcode.com/problems/minimum-one-bit-operations-to-make-integers-zero Title: 1611. Minimum One Bit Operations to Make Integers Zero Problem Description: Given an integer n, you must transform it into 0 using the following operations any number of times:

Change the rightmost (0th) bit in the binary representation of n. Change the ith bit in the binary representation of n if the (i-1)th bit is set to 1 and the (i-2)th through 0th bits are set to 0.

Return the minimum number of operations to transform n into 0. Example 1: Input: n = 3 Output: 2 Explanation: The binary representation of 3 is "11". "11" -> "01" with the 2nd operation since the 0th bit is 1. "01" -> "00" with the 1st operation.

Example 2: Input: n = 6 Output: 4 Explanation: The binary representation of 6 is "110". "110" -> "010" with the 2nd operation since the 1st bit is 1 and 0th through 0th bits are 0. "010" -> "011" with the 1st operation. "011" -> "001" with the 2nd operation since the 0th bit is 1. "001" -> "000" with the 1st operation.

Constraints:

0 <= n <= 109

==
Problem Number: 1942 URL: https://leetcode.com/problems/count-subtrees-

with-max-distance-between-cities Title: 1617. Count Subtrees With Max Distance Between Cities Problem Description: There are n cities numbered from 1 to n. You are given an array edges of size n-1, where edges[i] = [ui, vi] represents a bidirectional edge between cities ui and vi. There exists a unique path between each pair of cities. In other words, the cities form a tree. A subtree is a subset of cities where every city is reachable from every other city in the subset, where the path between each pair passes through only the cities from the subset. Two subtrees are different if there is a city in one subtree that is not present in the other. For each d from 1 to n-1, find the number of subtrees in which the maximum distance between any two cities in the subtree is equal to d. Return an array of size n-1 where the dth element (1-indexed) is the number of subtrees in which the maximum distance between any two cities is equal to d. Notice that the distance between the two cities is the number of edges in the path between them. Example 1:

Input: n = 4, edges = [[1,2],[2,3],[2,4]] Output: [3,4,0] Explanation: The subtrees with subsets {1,2}, {2,3} and {2,4} have a max distance of 1. The subtrees with subsets {1,2,3}, {1,2,4}, {2,3,4} and {1,2,3,4} have a max distance of 2. No subtree has two nodes where the max distance between them is 3.

Example 2: Input: n = 2, edges = [[1,2]] Output: [1]

Example 3: Input: n = 3, edges = [[1,2],[2,3]] Output: [2,1]

Constraints:

2 <= n <= 15 edges.length == n-1 edges[i].length == 2 1 <= ui, vi <= n All pairs (ui, vi) are distinct.

==
Problem Number: 1943 URL: https://leetcode.com/problems/fancy-sequence Title: 1622. Fancy Sequence Problem Description: Write an API that generates fancy sequences using the append, addAll, and multAll operations. Implement the Fancy class:

Fancy() Initializes the object with an empty sequence. void append(val) Appends an integer val to the end of the sequence. void addAll(inc) Increments all existing values in the sequence by an integer inc. void multAll(m) Multiplies all existing values in the sequence by an integer m. int getIndex(idx) Gets the current value at index idx (0-indexed) of the sequence modulo 109 + 7. If the index is greater or equal than the length of the sequence, return -1.

Example 1: Input ["Fancy", "append", "addAll", "append", "multAll", "getIndex", "addAll", "append", "multAll", "getIndex", "getIndex", "getIndex"] [[], [2], [3], [7], [2], [0], [3], [10], [2], [0], [1], [2]] Output [null, null, null, null, null, 10, null, null, null, 26, 34, 20]

Explanation Fancy fancy = new Fancy(); fancy.append(2); // fancy sequence: [2] fancy.addAll(3); // fancy sequence: [2+3] -> [5] fancy.append(7); // fancy sequence: [5, 7] fancy.multAll(2); // fancy sequence: [5*2, 7*2]

-> [10, 14] fancy.getIndex(0); // return 10 fancy.addAll(3); // fancy sequence: [10+3, 14+3] -> [13, 17] fancy.append(10); // fancy sequence: [13, 17, 10] fancy.multAll(2); // fancy sequence: [13*2, 17*2, 10*2] -> [26, 34, 20] fancy.getIndex(0); // return 26 fancy.getIndex(1); // return 34 fancy.getIndex(2); // return 20

Constraints:

1 <= val, inc, m <= 100 0 <= idx <= 105 At most 105 calls total will be made to append, addAll, multAll, and getIndex.

==
Problem Number: 1944 URL: https://leetcode.com/problems/graph-connectivity-with-threshold Title: 1627. Graph Connectivity With Threshold Problem Description: We have n cities labeled from 1 to n. Two different cities with labels x and y are directly connected by a bidirectional road if and only if x and y share a common divisor strictly greater than some threshold. More formally, cities with labels x and y have a road between them if there exists an integer z such that all of the following are true:

x % z == 0, y % z == 0, and z > threshold.

Given the two integers, n and threshold, and an array of queries, you must determine for each queries[i] = [ai, bi] if cities ai and bi are connected directly or indirectly. (i.e. there is some path between them). Return an array answer, where answer.length == queries.length and answer[i] is true if for the ith query, there is a path between ai and bi, or answer[i] is false if there is no path. Example 1:

Input: n = 6, threshold = 2, queries = [[1,4],[2,5],[3,6]] Output: [false,false,true] Explanation: The divisors for each number: 1: 1 2: 1, 2 3: 1, 3 4: 1, 2, 4 5: 1, 5 6: 1, 2, 3, 6 Using the underlined divisors above the threshold, only cities 3 and 6 share a common divisor, so they are the only ones directly connected. The result of each query: [1,4] 1 is not connected to 4 [2,5] 2 is not connected to 5 [3,6] 3 is connected to 6 through path 3--6

Example 2:

Input: n = 6, threshold = 0, queries = [[4,5],[3,4],[3,2],[2,6],[1,3]] Output: [true,true,true,true,true] Explanation: The divisors for each number are the same as the previous example. However, since the threshold is 0, all divisors can be used. Since all numbers share 1 as a divisor, all cities are connected.

Example 3:

Input: n = 5, threshold = 1, queries = [[4,5],[4,5],[3,2],[2,3],[3,4]] Output: [false,false,false,false,false] Explanation: Only cities 2 and 4 share a common divisor 2 which is strictly greater than the threshold 1, so they are the only ones directly connected. Please notice that there can be multiple queries for the same pair of nodes [x, y], and that the query [x, y] is equivalent to the query [y, x].

Constraints:

2 <= n <= 104 0 <= threshold <= n 1 <= queries.length <= 105 queries[i].length == 2 1 <= ai, bi <= cities ai != bi

==
Problem Number: 1945 URL: https://leetcode.com/problems/rank-transform-of-a-matrix Title: 1632. Rank Transform of a Matrix Problem Description: Given an m x n matrix, return a new matrix answer where answer[row][col] is the rank of matrix[row][col]. The rank is an integer that represents how large an element is compared to other elements. It is calculated using the following rules:

The rank is an integer starting from 1. If two elements p and q are in the same row or column, then:

If p < q then rank(p) < rank(q) If p == q then rank(p) == rank(q) If p > q then rank(p) > rank(q)

The rank should be as small as possible.

The test cases are generated so that answer is unique under the given rules. Example 1:

Input: matrix = [[1,2],[3,4]] Output: [[1,2],[2,3]] Explanation: The rank of matrix[0][0] is 1 because it is the smallest integer in its row and column. The rank of matrix[0][1] is 2 because matrix[0][1] > matrix[0][0] and matrix[0][0] is rank 1. The rank of matrix[1][0] is 2 because matrix[1][0] > matrix[0][0] and matrix[0][0] is rank 1. The rank of matrix[1][1] is 3 because matrix[1][1] > matrix[0][1], matrix[1][1] > matrix[1][0], and both matrix[0][1] and matrix[1][0] are rank 2.

Example 2:

Input: matrix = [[7,7],[7,7]] Output: [[1,1],[1,1]]

Example 3:

Input: matrix = [[20,-21,14],[-19,4,19],[22,-47,24],[-19,4,19]] Output: [[4,2,3],[1,3,4],[5,1,6],[1,3,4]]

Constraints:

m == matrix.length n == matrix[i].length 1 <= m, n <= 500 -109 <= matrix[row][col] <= 109

==
Problem Number: 1946 URL: https://leetcode.com/problems/number-of-ways-to-form-a-target-string-given-a-dictionary Title: 1639. Number of Ways to Form a Target String Given a Dictionary Problem Description: You are given a list of strings of the same length words and a string target. Your task is to form target using the given words under the following rules:

target should be formed from left to right. To form the ith character (0-indexed) of target, you can choose the kth character of the jth string in words if target[i] = words[j][k]. Once you use the kth character of the jth string of words, you can no longer use the xth character of any string in words where x <= k. In other words, all characters to the left of or at index k become unusuable for every string. Repeat the process until you form the string target.

Notice that you can use multiple characters from the same string in words provided the conditions above are met. Return the number of ways to form target from words. Since the answer may be too large, return it modulo 109 + 7. Example 1: Input: words = ["acca","bbbb","caca"], target = "aba" Output: 6 Explanation: There are 6 ways to form target. "aba" -> index 0 ("acca"), index 1 ("bbbb"), index 3 ("caca") "aba" -> index 0 ("acca"), index 2 ("bbbb"), index 3 ("caca") "aba" -> index 0 ("acca"), index 1 ("bbbb"), index 3 ("acca") "aba" -> index 0 ("acca"), index 2 ("bbbb"), index 3 ("acca") "aba" -> index 1 ("caca"), index 2 ("bbbb"), index 3 ("acca") "aba" -> index 1 ("caca"), index 2 ("bbbb"), index 3 ("caca")

Example 2: Input: words = ["abba","baab"], target = "bab" Output: 4 Explanation: There are 4 ways to form target. "bab" -> index 0 ("baab"), index 1 ("baab"), index 2 ("abba") "bab" -> index 0 ("baab"), index 1 ("baab"), index 3 ("baab") "bab" -> index 0 ("baab"), index 2 ("baab"), index 3 ("baab") "bab" -> index 1 ("abba"), index 2 ("baab"), index 3 ("baab")

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 1000 All strings in words have the same length. 1 <= target.length <= 1000 words[i] and target contain only lowercase English letters.

==

Problem Number: 1947 URL: https://leetcode.com/problems/kth-smallest-instructions Title: 1643. Kth Smallest Instructions Problem Description: Bob is standing at cell (0, 0), and he wants to reach destination: (row, column). He can only travel right and down. You are going to help Bob by providing instructions for him to reach destination. The instructions are represented as a string, where each character is either:

'H', meaning move horizontally (go right), or 'V', meaning move vertically (go down).

Multiple instructions will lead Bob to destination. For example, if destination is (2, 3), both "HHHVV" and "HVHVH" are valid instructions. However, Bob is very picky. Bob has a lucky number k, and he wants the kth lexicographically smallest instructions that will lead him to destination. k is 1-indexed. Given an integer array destination and an integer k, return the kth lexicographically smallest instructions that will take Bob to destination. Example 1:

Input: destination = [2,3], k = 1 Output: "HHHVV" Explanation: All the instructions that reach (2, 3) in lexicographic order are as follows: ["HHHVV",

”HHVHV”, ”HHVVH”, ”HVHHV”, ”HVHVH”, ”HVVHH”, ”VHHHV”, ”VH-HVH”, ”VHVHH”, ”VVHHH”].

Example 2:

Input: destination = [2,3], k = 2 Output: ”HHVHV”

Example 3:

Input: destination = [2,3], k = 3 Output: ”HHVVH”

Constraints:

destination.length == 2 1 <= row, column <= 15 1 <= k <= nCr(row + column, row), where nCr(a, b) denotes a choose b.

==

Problem Number: 1948 URL: https://leetcode.com/problems/create-sorted-array-through-instructions Title: 1649. Create Sorted Array through Instructions Problem Description: Given an integer array instructions, you are asked to create a sorted array from the elements in instructions. You start with an empty container nums. For each element from left to right in instructions, insert it into nums. The cost of each insertion is the minimum of the following:

The number of elements currently in nums that are strictly less than instructions[i]. The number of elements currently in nums that are strictly greater than instructions[i].

For example, if inserting element 3 into nums = [1,2,3,5], the cost of insertion is min(2, 1) (elements 1 and 2 are less than 3, element 5 is greater than 3) and nums will become [1,2,3,3,5]. Return the total cost to insert all elements from instructions into nums. Since the answer may be large, return it modulo 109 + 7 Example 1: Input: instructions = [1,5,6,2] Output: 1 Explanation: Begin with nums = []. Insert 1 with cost min(0, 0) = 0, now nums = [1]. Insert 5 with cost min(1, 0) = 0, now nums = [1,5]. Insert 6 with cost min(2, 0) = 0, now nums = [1,5,6]. Insert 2 with cost min(1, 2) = 1, now nums = [1,2,5,6]. The total cost is 0 + 0 + 0 + 1 = 1. Example 2: Input: instructions = [1,2,3,6,5,4] Output: 3 Explanation: Begin with nums = []. Insert 1 with cost min(0, 0) = 0, now nums = [1]. Insert 2 with cost min(1, 0) = 0, now nums = [1,2]. Insert 3 with cost min(2, 0) = 0, now nums = [1,2,3]. Insert 6 with cost min(3, 0) = 0, now nums = [1,2,3,6]. Insert 5 with cost min(3, 1) = 1, now nums = [1,2,3,5,6]. Insert 4 with cost min(3, 2) = 2, now nums = [1,2,3,4,5,6]. The total cost is 0 + 0 + 0 + 0 + 1 + 2 = 3.

Example 3: Input: instructions = [1,3,3,3,2,4,2,1,2] Output: 4 Explanation: Begin with nums = []. Insert 1 with cost min(0, 0) = 0, now nums = [1]. Insert 3 with cost min(1, 0) = 0, now nums = [1,3]. Insert 3 with cost min(1, 0) = 0, now nums = [1,3,3]. Insert 3 with cost min(1, 0) = 0, now nums = [1,3,3,3]. Insert 2 with cost min(1, 3) = 1, now nums = [1,2,3,3,3]. Insert 4 with cost min(5, 0) = 0, now nums = [1,2,3,3,3,4]. Insert 2 with cost min(1, 4) = 1, now nums = [1,2,2,3,3,3,4]. Insert 1 with cost min(0, 6) = 0, now nums =

[1,1,2,2,3,3,3,4]. Insert 2 with cost min(2, 4) = 2, now nums = [1,1,2,2,2,3,3,3,4]. The total cost is 0 + 0 + 0 + 0 + 1 + 0 + 1 + 0 + 2 = 4.

Constraints:

1 <= instructions.length <= 105 1 <= instructions[i] <= 105

==
Problem Number: 1949 URL: https://leetcode.com/problems/distribute-repeating-integers Title: 1655. Distribute Repeating Integers Problem Description: You are given an array of n integers, nums, where there are at most 50 unique values in the array. You are also given an array of m customer order quantities, quantity, where quantity[i] is the amount of integers the ith customer ordered. Determine if it is possible to distribute nums such that:

The ith customer gets exactly quantity[i] integers, The integers the ith customer gets are all equal, and Every customer is satisfied.

Return true if it is possible to distribute nums according to the above conditions. Example 1: Input: nums = [1,2,3,4], quantity = [2] Output: false Explanation: The 0th customer cannot be given two different integers.

Example 2: Input: nums = [1,2,3,3], quantity = [2] Output: true Explanation: The 0th customer is given [3,3]. The integers [1,2] are not used.

Example 3: Input: nums = [1,1,2,2], quantity = [2,2] Output: true Explanation: The 0th customer is given [1,1], and the 1st customer is given [2,2].

Constraints:

n == nums.length 1 <= n <= 105 1 <= nums[i] <= 1000 m == quantity.length 1 <= m <= 10 1 <= quantity[i] <= 105 There are at most 50 unique values in nums.

==
Problem Number: 1950 URL: https://leetcode.com/problems/maximize-grid-happiness Title: 1659. Maximize Grid Happiness Problem Description: You are given four integers, m, n, introvertsCount, and extrovertsCount. You have an m x n grid, and there are two types of people: introverts and extroverts. There are introvertsCount introverts and extrovertsCount extroverts. You should decide how many people you want to live in the grid and assign each of them one grid cell. Note that you do not have to have all the people living in the grid. The happiness of each person is calculated as follows:

Introverts start with 120 happiness and lose 30 happiness for each neighbor (introvert or extrovert). Extroverts start with 40 happiness and gain 20 happiness for each neighbor (introvert or extrovert).

Neighbors live in the directly adjacent cells north, east, south, and west of a person's cell. The grid happiness is the sum of each person's happiness. Return the maximum possible grid happiness. Example 1:

Input: m = 2, n = 3, introvertsCount = 1, extrovertsCount = 2 Output: 240 Explanation: Assume the grid is 1-indexed with coordinates (row, column). We can put the introvert in cell (1,1) and put the extroverts in cells (1,3) and (2,3). - Introvert at (1,1) happiness: 120 (starting happiness) - (0 * 30) (0 neighbors) = 120 - Extrovert at (1,3) happiness: 40 (starting happiness) + (1 * 20) (1 neighbor) = 60 - Extrovert at (2,3) happiness: 40 (starting happiness) + (1 * 20) (1 neighbor) = 60 The grid happiness is 120 + 60 + 60 = 240. The above figure shows the grid in this example with each person's happiness. The introvert stays in the light green cell while the extroverts live on the light purple cells.

Example 2: Input: m = 3, n = 1, introvertsCount = 2, extrovertsCount = 1 Output: 260 Explanation: Place the two introverts in (1,1) and (3,1) and the extrovert at (2,1). - Introvert at (1,1) happiness: 120 (starting happiness) - (1 * 30) (1 neighbor) = 90 - Extrovert at (2,1) happiness: 40 (starting happiness) + (2 * 20) (2 neighbors) = 80 - Introvert at (3,1) happiness: 120 (starting happiness) - (1 * 30) (1 neighbor) = 90 The grid happiness is 90 + 80 + 90 = 260.

Example 3: Input: m = 2, n = 2, introvertsCount = 4, extrovertsCount = 0 Output: 240

Constraints:

1 <= m, n <= 5 0 <= introvertsCount, extrovertsCount <= min(m * n, 6)

==
Problem Number: 1951 URL: https://leetcode.com/problems/minimum-initial-energy-to-finish-tasks Title: 1665. Minimum Initial Energy to Finish Tasks Problem Description: You are given an array tasks where tasks[i] = [actuali, minimumi]:

actuali is the actual amount of energy you spend to finish the ith task. minimumi is the minimum amount of energy you require to begin the ith task.

For example, if the task is [10, 12] and your current energy is 11, you cannot start this task. However, if your current energy is 13, you can complete this task, and your energy will be 3 after finishing it. You can finish the tasks in any order you like. Return the minimum initial amount of energy you will need to finish all the tasks. Example 1: Input: tasks = [[1,2],[2,4],[4,8]] Output: 8 Explanation: Starting with 8 energy, we finish the tasks in the following order: - 3rd task. Now energy = 8 - 4 = 4. - 2nd task. Now energy = 4 - 2 = 2. - 1st task. Now energy = 2 - 1 = 1. Notice that even though we have leftover energy, starting with 7 energy does not work because we cannot do the 3rd task. Example 2: Input: tasks = [[1,3],[2,4],[10,11],[10,12],[8,9]] Output: 32 Explanation: Starting with 32 energy, we finish the tasks in the following order: - 1st task. Now energy = 32 - 1 = 31. - 2nd task. Now energy = 31 - 2 = 29. - 3rd task. Now energy = 29 - 10 = 19. - 4th task. Now energy = 19 - 10 = 9. - 5th task. Now energy = 9 - 8 = 1. Example 3: Input: tasks = [[1,7],[2,8],[3,9],[4,10],[5,11],[6,12]] Output:

27 Explanation: Starting with 27 energy, we finish the tasks in the following order: - 5th task. Now energy = 27 - 5 = 22. - 2nd task. Now energy = 22 - 2 = 20. - 3rd task. Now energy = 20 - 3 = 17. - 1st task. Now energy = 17 - 1 = 16. - 4th task. Now energy = 16 - 4 = 12. - 6th task. Now energy = 12 - 6 = 6.

Constraints:

1 <= tasks.length <= 105 1 <= actuali <= minimumi <= 104

==

Problem Number: 1952 URL: https://leetcode.com/problems/minimum-number-of-removals-to-make-mountain-array Title: 1671. Minimum Number of Removals to Make Mountain Array Problem Description: You may recall that an array arr is a mountain array if and only if:

arr.length >= 3 There exists some index i (0-indexed) with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i] arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

Given an integer array nums, return the minimum number of elements to remove to make nums a mountain array. Example 1: Input: nums = [1,3,1] Output: 0 Explanation: The array itself is a mountain array so we do not need to remove any elements.

Example 2: Input: nums = [2,1,1,5,6,2,3,1] Output: 3 Explanation: One solution is to remove the elements at indices 0, 1, and 5, making the array nums = [1,5,6,3,1].

Constraints:

3 <= nums.length <= 1000 1 <= nums[i] <= 109 It is guaranteed that you can make a mountain array out of nums.

==

Problem Number: 1953 URL: https://leetcode.com/problems/minimize-deviation-in-array Title: 1675. Minimize Deviation in Array Problem Description: You are given an array nums of n positive integers. You can perform two types of operations on any element of the array any number of times:

If the element is even, divide it by 2.

For example, if the array is [1,2,3,4], then you can do this operation on the last element, and the array will be [1,2,3,2].

If the element is odd, multiply it by 2.

For example, if the array is [1,2,3,4], then you can do this operation on the first element, and the array will be [2,2,3,4].

The deviation of the array is the maximum difference between any two elements in the array. Return the minimum deviation the array can have after performing some number of operations. Example 1: Input: nums = [1,2,3,4] Output: 1 Explanation: You can transform the array to [1,2,3,2], then to [2,2,3,2], then the deviation will be 3 - 2 = 1.

Example 2: Input: nums = [4,1,5,20,3] Output: 3 Explanation: You can transform the array after two operations to [4,2,5,5,3], then the deviation will be 5 - 2 = 3.

Example 3: Input: nums = [2,10,8] Output: 3

Constraints:

n == nums.length 2 <= n <= 5 * 104 1 <= nums[i] <= 109

==

Problem Number: 1954 URL: https://leetcode.com/problems/minimum-incompatibility Title: 1681. Minimum Incompatibility Problem Description: You are given an integer array nums and an integer k. You are asked to distribute this array into k subsets of equal size such that there are no two equal elements in the same subset. A subset's incompatibility is the difference between the maximum and minimum elements in that array. Return the minimum possible sum of incompatibilities of the k subsets after distributing the array optimally, or return -1 if it is not possible. A subset is a group integers that appear in the array with no particular order. Example 1: Input: nums = [1,2,1,4], k = 2 Output: 4 Explanation: The optimal distribution of subsets is [1,2] and [1,4]. The incompatibility is (2-1) + (4-1) = 4. Note that [1,1] and [2,4] would result in a smaller sum, but the first subset contains 2 equal elements. Example 2: Input: nums = [6,3,8,1,3,1,2,2], k = 4 Output: 6 Explanation: The optimal distribution of subsets is [1,2], [2,3], [6,8], and [1,3]. The incompatibility is (2-1) + (3-2) + (8-6) + (3-1) = 6.

Example 3: Input: nums = [5,3,3,6,3,3], k = 3 Output: -1 Explanation: It is impossible to distribute nums into 3 subsets where no two elements are equal in the same subset.

Constraints:

1 <= k <= nums.length <= 16 nums.length is divisible by k 1 <= nums[i] <= nums.length

==

Problem Number: 1955 URL: https://leetcode.com/problems/delivering-boxes-from-storage-to-ports Title: 1687. Delivering Boxes from Storage to Ports Problem Description: You have the task of delivering some boxes from storage to their ports using only one ship. However, this ship has a limit on the number of boxes and the total weight that it can carry. You are given an array boxes, where boxes[i] = [portsi, weighti], and three integers portsCount, maxBoxes, and maxWeight.

portsi is the port where you need to deliver the ith box and weightsi is the weight of the ith box. portsCount is the number of ports. maxBoxes and maxWeight are the respective box and weight limits of the ship.

The boxes need to be delivered in the order they are given. The ship will follow these steps:

The ship will take some number of boxes from the boxes queue, not violating the maxBoxes and maxWeight constraints. For each loaded box in order, the ship will make a trip to the port the box needs to be delivered to and deliver it. If the ship is already at the correct port, no trip is needed, and the box can immediately be delivered. The ship then makes a return trip to storage to take more boxes from the queue.

The ship must end at storage after all the boxes have been delivered. Return the minimum number of trips the ship needs to make to deliver all boxes to their respective ports. Example 1: Input: boxes = [[1,1],[2,1],[1,1]], portsCount = 2, maxBoxes = 3, maxWeight = 3 Output: 4 Explanation: The optimal strategy is as follows: - The ship takes all the boxes in the queue, goes to port 1, then port 2, then port 1 again, then returns to storage. 4 trips. So the total number of trips is 4. Note that the first and third boxes cannot be delivered together because the boxes need to be delivered in order (i.e. the second box needs to be delivered at port 2 before the third box).

Example 2: Input: boxes = [[1,2],[3,3],[3,1],[3,1],[2,4]], portsCount = 3, maxBoxes = 3, maxWeight = 6 Output: 6 Explanation: The optimal strategy is as follows: - The ship takes the first box, goes to port 1, then returns to storage. 2 trips. - The ship takes the second, third and fourth boxes, goes to port 3, then returns to storage. 2 trips. - The ship takes the fifth box, goes to port 2, then returns to storage. 2 trips. So the total number of trips is 2 + 2 + 2 = 6.

Example 3: Input: boxes = [[1,4],[1,2],[2,1],[2,1],[3,2],[3,4]], portsCount = 3, maxBoxes = 6, maxWeight = 7 Output: 6 Explanation: The optimal strategy is as follows: - The ship takes the first and second boxes, goes to port 1, then returns to storage. 2 trips. - The ship takes the third and fourth boxes, goes to port 2, then returns to storage. 2 trips. - The ship takes the fifth and sixth boxes, goes to port 3, then returns to storage. 2 trips. So the total number of trips is 2 + 2 + 2 = 6.

Constraints:

1 <= boxes.length <= 105 1 <= portsCount, maxBoxes, maxWeight <= 105 1 <= portsi <= portsCount 1 <= weightsi <= maxWeight

== Problem Number: 1956 URL: https://leetcode.com/problems/maximum-height-by-stacking-cuboids Title: 1691. Maximum Height by Stacking Cuboids Problem Description: Given n cuboids where the dimensions of the ith cuboid is cuboids[i] = [widthi, lengthi, heighti] (0-indexed). Choose a subset of cuboids and place them on each other. You can place cuboid i on cuboid j if widthi <= widthj and lengthi <= lengthj and heighti <= heightj. You can rearrange any cuboid's dimensions by rotating it to put it on another cuboid. Return the maximum height of the stacked cuboids.   Example 1:

Input: cuboids = [[50,45,20],[95,37,53],[45,23,12]] Output: 190 Explanation: Cuboid 1 is placed on the bottom with the 53x37 side facing down with height 95. Cuboid 0 is placed next with the 45x20 side facing down with height 50. Cuboid 2 is placed next with the 23x12 side facing down with height 45. The total height is 95 + 50 + 45 = 190.

Example 2: Input: cuboids = [[38,25,45],[76,35,3]] Output: 76 Explanation: You can't place any of the cuboids on the other. We choose cuboid 1 and rotate it so that the 35x3 side is facing down and its height is 76.

Example 3: Input: cuboids = [[7,11,17],[7,17,11],[11,7,17],[11,17,7],[17,7,11],[17,11,7]] Output: 102 Explanation: After rearranging the cuboids, you can see that all cuboids have the same dimension. You can place the 11x7 side down on all cuboids so their heights are 17. The maximum height of stacked cuboids is 6 * 17 = 102.

Constraints:

n == cuboids.length 1 <= n <= 100 1 <= widthi, lengthi, heighti <= 100

==
Problem Number: 1957 URL: https://leetcode.com/problems/checking-existence-of-edge-length-limited-paths Title: 1697. Checking Existence of Edge Length Limited Paths Problem Description: An undirected graph of n nodes is defined by edgeList, where edgeList[i] = [ui, vi, disi] denotes an edge between nodes ui and vi with distance disi. Note that there may be multiple edges between two nodes. Given an array queries, where queries[j] = [pj, qj, limitj], your task is to determine for each queries[j] whether there is a path between pj and qj such that each edge on the path has a distance strictly less than limitj . Return a boolean array answer, where answer.length == queries.length and the jth value of answer is true if there is a path for queries[j] is true, and false otherwise.   Example 1:

Input: n = 3, edgeList = [[0,1,2],[1,2,4],[2,0,8],[1,0,16]], queries = [[0,1,2],[0,2,5]] Output: [false,true] Explanation: The above figure shows the given graph. Note that there are two overlapping edges between 0 and 1 with distances 2 and 16. For the first query, between 0 and 1 there is no path where each distance is less than 2, thus we return false for this query. For the second query, there is a path (0 -> 1 -> 2) of two edges with distances less than 5, thus we return true for this query.

Example 2:

Input: n = 5, edgeList = [[0,1,10],[1,2,5],[2,3,9],[3,4,13]], queries = [[0,4,14],[1,4,13]] Output: [true,false] Explanation: The above figure shows the given graph.

Constraints:

2 <= n <= 105 1 <= edgeList.length, queries.length <= 105 edgeList[i].length == 3 queries[j].length == 3 0 <= ui, vi, pj, qj <= n - 1 ui != vi pj != qj 1 <= disi, limitj <= 109 There may be multiple edges between two nodes.

==
Problem Number: 1958 URL: https://leetcode.com/problems/minimum-adjacent-swaps-for-k-consecutive-ones Title: 1703. Minimum Adjacent Swaps for K Consecutive Ones Problem Description: You are given an integer array, nums, and an integer k. nums comprises of only 0's and 1's. In one move, you can choose two adjacent indices and swap their values. Return the minimum number of moves required so that nums has k consecutive 1's. Example 1: Input: nums = [1,0,0,1,0,1], k = 2 Output: 1 Explanation: In 1 move, nums could be [1,0,0,0,1,1] and have 2 consecutive 1's.

Example 2: Input: nums = [1,0,0,0,0,0,1,1], k = 3 Output: 5 Explanation: In 5 moves, the leftmost 1 can be shifted right until nums = [0,0,0,0,0,1,1,1].

Example 3: Input: nums = [1,1,0,1], k = 2 Output: 0 Explanation: nums already has 2 consecutive 1's.

Constraints:

1 <= nums.length <= 105 nums[i] is 0 or 1. 1 <= k <= sum(nums)

==
Problem Number: 1959 URL: https://leetcode.com/problems/maximum-xor-with-an-element-from-array Title: 1707. Maximum XOR With an Element From Array Problem Description: You are given an array nums consisting of non-negative integers. You are also given a queries array, where queries[i] = [xi, mi]. The answer to the ith query is the maximum bitwise XOR value of xi and any element of nums that does not exceed mi. In other words, the answer is max(nums[j] XOR xi) for all j such that nums[j] <= mi. If all elements in nums are larger than mi, then the answer is -1. Return an integer array answer where answer.length == queries.length and answer[i] is the answer to the ith query. Example 1: Input: nums = [0,1,2,3,4], queries = [[3,1],[1,3],[5,6]] Output: [3,3,7] Explanation: 1) 0 and 1 are the only two integers not greater than 1. 0 XOR 3 = 3 and 1 XOR 3 = 2. The larger of the two is 3. 2) 1 XOR 2 = 3. 3) 5 XOR 2 = 7.

Example 2: Input: nums = [5,2,4,6,6,3], queries = [[12,4],[8,1],[6,3]] Output: [15,-1,5]

Constraints:

1 <= nums.length, queries.length <= 105 queries[i].length == 2 0 <= nums[j], xi, mi <= 109

==
Problem Number: 1960 URL: https://leetcode.com/problems/minimum-operations-to-make-a-subsequence Title: 1713. Minimum Operations to Make a Subsequence Problem Description: You are given an array target that consists of distinct integers and another integer array arr that can have duplicates. In one operation, you can insert any integer at any position in arr. For example, if arr = [1,4,1,2], you can add 3 in the middle and make it [1,4,3,1,2]. Note that you can insert the integer at the very beginning or end of the array. Return the minimum number of operations needed to make target a subsequence of arr. A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order. For example, [2,7,4] is a subsequence of [4,2,3,7,2,1,4] (the underlined elements), while [2,4,2] is not. Example 1: Input: target = [5,1,3], arr = [9,4,2,3,4] Output: 2 Explanation: You can add 5 and 1 in such a way that makes arr = [5,9,4,1,2,3,4], then target will be a subsequence of arr.

Example 2: Input: target = [6,4,8,1,3,2], arr = [4,7,6,2,3,8,6,1] Output: 3

Constraints:

1 <= target.length, arr.length <= 105 1 <= target[i], arr[i] <= 109 target contains no duplicates.

==
Problem Number: 1961 URL: https://leetcode.com/problems/number-of-ways-to-reconstruct-a-tree Title: 1719. Number Of Ways To Reconstruct A Tree Problem Description: You are given an array pairs, where pairs[i] = [xi, yi], and:

There are no duplicates. xi < yi

Let ways be the number of rooted trees that satisfy the following conditions:

The tree consists of nodes whose values appeared in pairs. A pair [xi, yi] exists in pairs if and only if xi is an ancestor of yi or yi is an ancestor of xi. Note: the tree does not have to be a binary tree.

Two ways are considered to be different if there is at least one node that has different parents in both ways. Return:

0 if ways == 0 1 if ways == 1 2 if ways > 1

A rooted tree is a tree that has a single root node, and all edges are oriented to be outgoing from the root. An ancestor of a node is any node on the path from the root to that node (excluding the node itself). The root has no ancestors. Example 1:

Input: pairs = [[1,2],[2,3]] Output: 1 Explanation: There is exactly one valid rooted tree, which is shown in the above figure.

Example 2:

Input: pairs = [[1,2],[2,3],[1,3]] Output: 2 Explanation: There are multiple valid rooted trees. Three of them are shown in the above figures.

Example 3: Input: pairs = [[1,2],[2,3],[2,4],[1,5]] Output: 0 Explanation: There are no valid rooted trees.   Constraints:

1 <= pairs.length <= 105 1 <= xi < yi <= 500 The elements in pairs are unique.

==
Problem Number: 1962 URL: https://leetcode.com/problems/find-minimum-time-to-finish-all-jobs Title: 1723. Find Minimum Time to Finish All Jobs Problem Description: You are given an integer array jobs, where jobs[i] is the amount of time it takes to complete the ith job. There are k workers that you can assign jobs to. Each job should be assigned to exactly one worker. The working time of a worker is the sum of the time it takes to complete all jobs assigned to them. Your goal is to devise an optimal assignment such that the maximum working time of any worker is minimized. Return the minimum possible maximum working time of any assignment.   Example 1: Input: jobs = [3,2,3], k = 3 Output: 3 Explanation: By assigning each person one job, the maximum time is 3.

Example 2: Input: jobs = [1,2,4,7,8], k = 2 Output: 11 Explanation: Assign the jobs the following way: Worker 1: 1, 2, 8 (working time = 1 + 2 + 8 = 11) Worker 2: 4, 7 (working time = 4 + 7 = 11) The maximum working time is 11.   Constraints:

1 <= k <= jobs.length <= 12 1 <= jobs[i] <= 107

==
Problem Number: 1963 URL: https://leetcode.com/problems/cat-and-mouse-ii Title: 1728. Cat and Mouse II Problem Description: A game is played by a cat and a mouse named Cat and Mouse. The environment is represented by a grid of size rows x cols, where each element is a wall, floor, player (Cat, Mouse), or food.

Players are represented by the characters 'C'(Cat),'M'(Mouse). Floors are represented by the character '.' and can be walked on. Walls are represented by the character '#' and cannot be walked on. Food is represented by the character 'F' and can be walked on. There is only one of each character 'C', 'M', and 'F' in grid.

Mouse and Cat play according to the following rules:

Mouse moves first, then they take turns to move. During each turn, Cat and Mouse can jump in one of the four directions (left, right, up, down). They cannot jump over the wall nor outside of the grid. catJump, mouseJump are the maximum lengths Cat and Mouse can jump at a time, respectively. Cat and Mouse can jump less than the maximum length. Staying in the same position is allowed. Mouse can jump over Cat.

The game can end in 4 ways:

If Cat occupies the same position as Mouse, Cat wins. If Cat reaches the food first, Cat wins. If Mouse reaches the food first, Mouse wins. If Mouse cannot get to the food within 1000 turns, Cat wins.

Given a rows x cols matrix grid and two integers catJump and mouseJump, return true if Mouse can win the game if both Cat and Mouse play optimally, otherwise return false. Example 1:

Input: grid = ["####F","#C...","M...."], catJump = 1, mouseJump = 2 Output: true Explanation: Cat cannot catch Mouse on its turn nor can it get the food before Mouse.

Example 2:

Input: grid = ["M.C...F"], catJump = 1, mouseJump = 4 Output: true

Example 3: Input: grid = ["M.C...F"], catJump = 1, mouseJump = 3 Output: false

Constraints:

rows == grid.length cols = grid[i].length 1 <= rows, cols <= 8 grid[i][j] consist only of characters 'C', 'M', 'F', '.', and '#'. There is only one of each character 'C', 'M', and 'F' in grid. 1 <= catJump, mouseJump <= 8

==
Problem Number: 1964 URL: https://leetcode.com/problems/count-ways-to-make-array-with-product Title: 1735. Count Ways to Make Array With Product Problem Description: You are given a 2D integer array, queries. For each queries[i], where queries[i] = [ni, ki], find the number of different ways you can place positive integers into an array of size ni such that the product of the integers is ki. As the number of ways may be too large, the answer to the ith query is the number of ways modulo 109 + 7. Return an integer array answer where answer.length == queries.length, and answer[i] is the answer to the ith query. Example 1: Input: queries = [[2,6],[5,1],[73,660]] Output: [4,1,50734910] Explanation: Each query is independent. [2,6]: There are 4 ways to fill an array of size 2 that multiply to 6: [1,6], [2,3], [3,2], [6,1]. [5,1]: There is 1 way to fill an array of size 5 that multiply to 1: [1,1,1,1,1]. [73,660]: There are 1050734917 ways to fill an array of size 73 that multiply to 660. 1050734917 modulo 109 + 7 = 50734910.

Example 2: Input: queries = [[1,1],[2,2],[3,3],[4,4],[5,5]] Output: [1,2,3,10,5]

Constraints:

1 <= queries.length <= 104 1 <= ni, ki <= 104

==
Problem Number: 1965 URL: https://leetcode.com/problems/building-boxes Title: 1739. Building Boxes Problem Description: You have a cubic storeroom

where the width, length, and height of the room are all equal to n units. You are asked to place n boxes in this room where each box is a cube of unit side length. There are however some rules to placing the boxes:

You can place the boxes anywhere on the floor. If box x is placed on top of the box y, then each side of the four vertical sides of the box y must either be adjacent to another box or to a wall.

Given an integer n, return the minimum possible number of boxes touching the floor.   Example 1:

Input: n = 3 Output: 3 Explanation: The figure above is for the placement of the three boxes. These boxes are placed in the corner of the room, where the corner is on the left side.

Example 2:

Input: n = 4 Output: 3 Explanation: The figure above is for the placement of the four boxes. These boxes are placed in the corner of the room, where the corner is on the left side.

Example 3:

Input: n = 10 Output: 6 Explanation: The figure above is for the placement of the ten boxes. These boxes are placed in the corner of the room, where the corner is on the back side.   Constraints:

1 <= n <= 109

==

Problem Number: 1966 URL: https://leetcode.com/problems/palindrome-partitioning-iv Title: 1745. Palindrome Partitioning IV Problem Description: Given a string s, return true if it is possible to split the string s into three non-empty palindromic substrings. Otherwise, return false. A string is said to be palindrome if it the same string when reversed.   Example 1: Input: s = ”abcbdd” Output: true Explanation: ”abcbdd” = ”a” + ”bcb” + ”dd”, and all three substrings are palindromes.

Example 2: Input: s = ”bcbddxy” Output: false Explanation: s cannot be split into 3 palindromes.

Constraints:

3 <= s.length <= 2000 s consists only of lowercase English letters.

==

Problem Number: 1967 URL: https://leetcode.com/problems/maximum-number-of-events-that-can-be-attended-ii Title: 1751. Maximum Number of Events That Can Be Attended II Problem Description: You are given an array of events where events[i] = [startDayi, endDayi, valuei]. The ith event starts at startDayi and ends at endDayi, and if you attend this event, you will receive a value of valuei. You are also given an integer k which represents the maximum number of events you can attend. You can only attend one event at a time. If you choose to attend an event, you must attend the entire event. Note that the end day is inclusive: that is, you cannot attend two events where one of them starts and the other ends on the same day. Return the maximum sum of values that you can receive by attending events.   Example 1:

Input: events = [[1,2,4],[3,4,3],[2,3,1]], k = 2 Output: 7 Explanation: Choose the green events, 0 and 1 (0-indexed) for a total value of 4 + 3 = 7. Example 2:

Input: events = [[1,2,4],[3,4,3],[2,3,10]], k = 2 Output: 10 Explanation: Choose event 2 for a total value of 10. Notice that you cannot attend any other event as they overlap, and that you do not have to attend k events. Example 3:

Input: events = [[1,1,1],[2,2,2],[3,3,3],[4,4,4]], k = 3 Output: 9 Explanation: Although the events do not overlap, you can only attend 3 events. Pick the highest valued three.   Constraints:

1 <= k <= events.length 1 <= k * events.length <= 106 1 <= startDayi <= endDayi <= 109 1 <= valuei <= 106

==
Problem Number: 1968 URL: https://leetcode.com/problems/closest-subsequence-sum Title: 1755. Closest Subsequence Sum Problem Description: You are given an integer array nums and an integer goal. You want to choose a subsequence of nums such that the sum of its elements is the closest possible to goal. That is, if the sum of the subsequence's elements is sum, then you want to minimize the absolute difference abs(sum - goal). Return the minimum possible value of abs(sum - goal). Note that a subsequence of an array is an array formed by removing some elements (possibly all or none) of the original array.   Example 1: Input: nums = [5,-7,3,5], goal = 6 Output: 0 Explanation: Choose the whole array as a subsequence, with a sum of 6. This is equal to the goal, so the absolute difference is 0.

Example 2: Input: nums = [7,-9,15,-2], goal = -5 Output: 1 Explanation: Choose the subsequence [7,-9,-2], with a sum of -4. The absolute difference is abs(-4 - (-5)) = abs(1) = 1, which is the minimum.

Example 3: Input: nums = [1,2,3], goal = -7 Output: 7

Constraints:

1 <= nums.length <= 40 -107 <= nums[i] <= 107 -109 <= goal <= 109

==
Problem Number: 1969 URL: https://leetcode.com/problems/minimum-degree-of-a-connected-trio-in-a-graph Title: 1761. Minimum Degree of a Connected Trio in a Graph Problem Description: You are given an undirected graph. You are given an integer n which is the number of nodes in the graph and an array edges, where each edges[i] = [ui, vi] indicates that there is an undirected edge between ui and vi. A connected trio is a set of three nodes where there is an edge between every pair of them. The degree of a connected trio is the number

of edges where one endpoint is in the trio, and the other is not. Return the minimum degree of a connected trio in the graph, or -1 if the graph has no connected trios.   Example 1:

Input: n = 6, edges = [[1,2],[1,3],[3,2],[4,1],[5,2],[3,6]] Output: 3 Explanation: There is exactly one trio, which is [1,2,3]. The edges that form its degree are bolded in the figure above.

Example 2:

Input: n = 7, edges = [[1,3],[4,1],[4,3],[2,5],[5,6],[6,7],[7,5],[2,6]] Output: 0 Explanation: There are exactly three trios: 1) [1,4,3] with degree 0. 2) [2,5,6] with degree 2. 3) [5,6,7] with degree 2.

Constraints:

2 <= n <= 400 edges[i].length == 2 1 <= edges.length <= n * (n-1) / 2 1 <= ui, vi <= n ui != vi There are no repeated edges.

==
Problem Number: 1970 URL: https://leetcode.com/problems/tree-of-coprimes Title: 1766. Tree of Coprimes Problem Description: There is a tree (i.e., a connected, undirected graph that has no cycles) consisting of n nodes numbered from 0 to n - 1 and exactly n - 1 edges. Each node has a value associated with it, and the root of the tree is node 0. To represent this tree, you are given an integer array nums and a 2D array edges. Each nums[i] represents the ith node's value, and each edges[j] = [uj, vj] represents an edge between nodes uj and vj in the tree. Two values x and y are coprime if gcd(x, y) == 1 where gcd(x, y) is the greatest common divisor of x and y. An ancestor of a node i is any other node on the shortest path from node i to the root. A node is not considered an ancestor of itself. Return an array ans of size n, where ans[i] is the closest ancestor to node i such that nums[i] and nums[ans[i]] are coprime, or -1 if there is no such ancestor.   Example 1:

Input: nums = [2,3,3,2], edges = [[0,1],[1,2],[1,3]] Output: [-1,0,0,1] Explanation: In the above figure, each node's value is in parentheses. - Node 0 has no coprime ancestors. - Node 1 has only one ancestor, node 0. Their values are coprime (gcd(2,3) == 1). - Node 2 has two ancestors, nodes 1 and 0. Node 1's value is not coprime (gcd(3,3) == 3), but node 0's value is (gcd(2,3) == 1), so node 0 is the closest valid ancestor. - Node 3 has two ancestors, nodes 1 and 0. It is coprime with node 1 (gcd(3,2) == 1), so node 1 is its closest valid ancestor.

Example 2:

Input: nums = [5,6,10,2,3,6,15], edges = [[0,1],[0,2],[1,3],[1,4],[2,5],[2,6]] Output: [-1,0,-1,0,0,0,-1]

Constraints:

nums.length == n 1 <= nums[i] <= 50 1 <= n <= 105 edges.length == n - 1 edges[j].length == 2 0 <= uj, vj < n uj != vj

==

Problem Number: 1971 URL: https://leetcode.com/problems/maximum-score-from-performing-multiplication-operations Title: 1770. Maximum Score from Performing Multiplication Operations Problem Description: You are given two 0-indexed integer arrays nums and multipliers of size n and m respectively, where n >= m. You begin with a score of 0. You want to perform exactly m operations. On the ith operation (0-indexed) you will:

Choose one integer x from either the start or the end of the array nums. Add multipliers[i] * x to your score.

Note that multipliers[0] corresponds to the first operation, multipliers[1] to the second operation, and so on.

Remove x from nums.

Return the maximum score after performing m operations. Example 1: Input: nums = [1,2,3], multipliers = [3,2,1] Output: 14 Explanation: An optimal solution is as follows: - Choose from the end, [1,2,3], adding 3 * 3 = 9 to the score. - Choose from the end, [1,2], adding 2 * 2 = 4 to the score. - Choose from the end, [1], adding 1 * 1 = 1 to the score. The total score is 9 + 4 + 1 = 14. Example 2: Input: nums = [-5,-3,-3,-2,7,1], multipliers = [-10,-5,3,4,6] Output: 102 Explanation: An optimal solution is as follows: - Choose from the start, [-5,-3,-3,-2,7,1], adding -5 * -10 = 50 to the score. - Choose from the start, [-3,-3,-2,7,1], adding -3 * -5 = 15 to the score. - Choose from the start, [-3,-2,7,1], adding -3 * 3 = -9 to the score. - Choose from the end, [-2,7,1], adding 1 * 4 = 4 to the score. - Choose from the end, [-2,7], adding 7 * 6 = 42 to the score. The total score is 50 + 15 - 9 + 4 + 42 = 102.

Constraints:

n == nums.length m == multipliers.length 1 <= m <= 300 m <= n <= 105 -1000 <= nums[i], multipliers[i] <= 1000

==

Problem Number: 1972 URL: https://leetcode.com/problems/maximize-palindrome-length-from-subsequences Title: 1771. Maximize Palindrome Length From Subsequences Problem Description: You are given two strings, word1 and word2. You want to construct a string in the following manner:

Choose some non-empty subsequence subsequence1 from word1. Choose some non-empty subsequence subsequence2 from word2. Concatenate the subsequences: subsequence1 + subsequence2, to make the string.

Return the length of the longest palindrome that can be constructed in the described manner. If no palindromes can be constructed, return 0. A subsequence of a string s is a string that can be made by deleting some (possibly none) characters from s without changing the order of the remaining characters. A palindrome is a string that reads the same forward as well as backward. Example 1: Input: word1 = "cacb", word2 = "cbba" Output: 5 Explanation:

Choose "ab" from word1 and "cba" from word2 to make "abcba", which is a palindrome. Example 2: Input: word1 = "ab", word2 = "ab" Output: 3 Explanation: Choose "ab" from word1 and "a" from word2 to make "aba", which is a palindrome. Example 3: Input: word1 = "aa", word2 = "bb" Output: 0 Explanation: You cannot construct a palindrome from the described method, so return 0. Constraints:

1 <= word1.length, word2.length <= 1000 word1 and word2 consist of lowercase English letters.

==
Problem Number: 1973 URL: https://leetcode.com/problems/car-fleet-ii Title: 1776. Car Fleet II Problem Description: There are n cars traveling at different speeds in the same direction along a one-lane road. You are given an array cars of length n, where cars[i] = [positioni, speedi] represents:

positioni is the distance between the ith car and the beginning of the road in meters. It is guaranteed that positioni < positioni+1. speedi is the initial speed of the ith car in meters per second.

For simplicity, cars can be considered as points moving along the number line. Two cars collide when they occupy the same position. Once a car collides with another car, they unite and form a single car fleet. The cars in the formed fleet will have the same position and the same speed, which is the initial speed of the slowest car in the fleet. Return an array answer, where answer[i] is the time, in seconds, at which the ith car collides with the next car, or -1 if the car does not collide with the next car. Answers within 10-5 of the actual answers are accepted. Example 1: Input: cars = [[1,2],[2,1],[4,3],[7,2]] Output: [1.00000,-1.00000,3.00000,-1.00000] Explanation: After exactly one second, the first car will collide with the second car, and form a car fleet with speed 1 m/s. After exactly 3 seconds, the third car will collide with the fourth car, and form a car fleet with speed 2 m/s.

Example 2: Input: cars = [[3,4],[5,4],[6,3],[9,1]] Output: [2.00000,1.00000,1.50000,-1.00000]

Constraints:

1 <= cars.length <= 105 1 <= positioni, speedi <= 106 positioni < positioni+1

==
Problem Number: 1974 URL: https://leetcode.com/problems/count-pairs-of-nodes Title: 1782. Count Pairs Of Nodes Problem Description: You are given an undirected graph defined by an integer n, the number of nodes, and a 2D integer array edges, the edges in the graph, where edges[i] = [ui, vi] indicates that there is an undirected edge between ui and vi. You are also given an integer array queries. Let incident(a, b) be defined as the number of edges that are connected to either node a or b. The answer to the jth query is the number of pairs of nodes (a, b) that satisfy both of the following conditions:

a < b incident(a, b) > queries[j]

Return an array answers such that answers.length == queries.length and answers[j] is the answer of the jth query. Note that there can be multiple edges between the same two nodes.   Example 1:

Input: n = 4, edges = [[1,2],[2,4],[1,3],[2,3],[2,1]], queries = [2,3] Output: [6,5] Explanation: The calculations for incident(a, b) are shown in the table above. The answers for each of the queries are as follows: - answers[0] = 6. All the pairs have an incident(a, b) value greater than 2. - answers[1] = 5. All the pairs except (3, 4) have an incident(a, b) value greater than 3.

Example 2: Input: n = 5, edges = [[1,5],[1,5],[3,4],[2,5],[1,3],[5,1],[2,3],[2,5]], queries = [1,2,3,4,5] Output: [10,10,9,8,6]

Constraints:

2 <= n <= 2 * 104 1 <= edges.length <= 105 1 <= ui, vi <= n ui != vi 1 <= queries.length <= 20 0 <= queries[j] < edges.length

==
Problem Number: 1975 URL: https://leetcode.com/problems/make-the-xor-of-all-segments-equal-to-zero Title: 1787. Make the XOR of All Segments Equal to Zero Problem Description: You are given an array nums and an integer k . The XOR of a segment [left, right] where left <= right is the XOR of all the elements with indices between left and right, inclusive: nums[left] XOR nums[left+1] XOR ... XOR nums[right]. Return the minimum number of elements to change in the array such that the XOR of all segments of size k is equal to zero.   Example 1: Input: nums = [1,2,0,3,0], k = 1 Output: 3 Explanation: Modify the array from [1,2,0,3,0] to from [0,0,0,0,0].

Example 2: Input: nums = [3,4,5,2,1,7,3,4,7], k = 3 Output: 3 Explanation: Modify the array from [3,4,5,2,1,7,3,4,7] to [3,4,7,3,4,7,3,4,7].

Example 3: Input: nums = [1,2,4,1,2,5,1,2,6], k = 3 Output: 3 Explanation: Modify the array from [1,2,4,1,2,5,1,2,6] to [1,2,3,1,2,3,1,2,3].   Constraints:

1 <= k <= nums.length <= 2000 0 <= nums[i] < 210

==
Problem Number: 1976 URL: https://leetcode.com/problems/maximum-score-of-a-good-subarray Title: 1793. Maximum Score of a Good Subarray Problem Description: You are given an array of integers nums (0-indexed) and an integer k. The score of a subarray (i, j) is defined as min(nums[i], nums[i+1], ..., nums[j]) * (j - i + 1). A good subarray is a subarray where i <= k <= j. Return the maximum possible score of a good subarray.   Example 1: Input: nums = [1,4,3,7,4,5], k = 3 Output: 15 Explanation: The optimal subarray is (1, 5) with a score of min(4,3,7,4,5) * (5-1+1) = 3 * 5 = 15.

Example 2: Input: nums = [5,5,4,5,4,1,1,1], k = 0 Output: 20 Explanation: The optimal subarray is (0, 4) with a score of min(5,5,4,5,4) * (4-0+1) = 4 * 5

= 20.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 2 * 104 0 <= k < nums.length

==
Problem Number: 1977 URL: https://leetcode.com/problems/maximize-score-after-n-operations Title: 1799. Maximize Score After N Operations Problem Description: You are given nums, an array of positive integers of size 2 * n. You must perform n operations on this array. In the ith operation (1-indexed), you will:

Choose two elements, x and y. Receive a score of i * gcd(x, y). Remove x and y from nums.

Return the maximum score you can receive after performing n operations. The function gcd(x, y) is the greatest common divisor of x and y. Example 1: Input: nums = [1,2] Output: 1 Explanation: The optimal choice of operations is: (1 * gcd(1, 2)) = 1

Example 2: Input: nums = [3,4,6,8] Output: 11 Explanation: The optimal choice of operations is: (1 * gcd(3, 6)) + (2 * gcd(4, 8)) = 3 + 8 = 11

Example 3: Input: nums = [1,2,3,4,5,6] Output: 14 Explanation: The optimal choice of operations is: (1 * gcd(1, 5)) + (2 * gcd(2, 4)) + (3 * gcd(3, 6)) = 1 + 4 + 9 = 14

Constraints:

1 <= n <= 7 nums.length == 2 * n 1 <= nums[i] <= 106

==
Problem Number: 1978 URL: https://leetcode.com/problems/count-pairs-with-xor-in-a-range Title: 1803. Count Pairs With XOR in a Range Problem Description: Given a (0-indexed) integer array nums and two integers low and high, return the number of nice pairs. A nice pair is a pair (i, j) where 0 <= i < j < nums.length and low <= (nums[i] XOR nums[j]) <= high. Example 1: Input: nums = [1,4,2,7], low = 2, high = 6 Output: 6 Explanation: All nice pairs (i, j) are as follows: - (0, 1): nums[0] XOR nums[1] = 5 - (0, 2): nums[0] XOR nums[2] = 3 - (0, 3): nums[0] XOR nums[3] = 6 - (1, 2): nums[1] XOR nums[2] = 6 - (1, 3): nums[1] XOR nums[3] = 3 - (2, 3): nums[2] XOR nums[3] = 5

Example 2: Input: nums = [9,8,4,2,1], low = 5, high = 14 Output: 8 Explanation: All nice pairs (i, j) are as follows: - (0, 2): nums[0] XOR nums[2] = 13 - (0, 3): nums[0] XOR nums[3] = 11 - (0, 4): nums[0] XOR nums[4] = 8 - (1, 2): nums[1] XOR nums[2] = 12 - (1, 3): nums[1] XOR nums[3] = 10 - (1, 4): nums[1] XOR nums[4] = 9 - (2, 3): nums[2] XOR nums[3] = 6 - (2, 4): nums[2] XOR nums[4] = 5 Constraints:

1 <= nums.length <= 2 * 104 1 <= nums[i] <= 2 * 104 1 <= low <= high <= 2 * 104

==
Problem Number: 1979 URL: https://leetcode.com/problems/maximize-number-of-nice-divisors Title: 1808. Maximize Number of Nice Divisors Problem Description: You are given a positive integer primeFactors. You are asked to construct a positive integer n that satisfies the following conditions:

The number of prime factors of n (not necessarily distinct) is at most primeFactors. The number of nice divisors of n is maximized. Note that a divisor of n is nice if it is divisible by every prime factor of n. For example, if n = 12, then its prime factors are [2,2,3], then 6 and 12 are nice divisors, while 3 and 4 are not.

Return the number of nice divisors of n. Since that number can be too large, return it modulo 109 + 7. Note that a prime number is a natural number greater than 1 that is not a product of two smaller natural numbers. The prime factors of a number n is a list of prime numbers such that their product equals n. Example 1: Input: primeFactors = 5 Output: 6 Explanation: 200 is a valid value of n. It has 5 prime factors: [2,2,2,5,5], and it has 6 nice divisors: [10,20,40,50,100,200]. There is not other value of n that has at most 5 prime factors and more nice divisors.

Example 2: Input: primeFactors = 8 Output: 18

Constraints:

1 <= primeFactors <= 109

==
Problem Number: 1980 URL: https://leetcode.com/problems/maximum-number-of-groups-getting-fresh-donuts Title: 1815. Maximum Number of Groups Getting Fresh Donuts Problem Description: There is a donuts shop that bakes donuts in batches of batchSize. They have a rule where they must serve all of the donuts of a batch before serving any donuts of the next batch. You are given an integer batchSize and an integer array groups, where groups[i] denotes that there is a group of groups[i] customers that will visit the shop. Each customer will get exactly one donut. When a group visits the shop, all customers of the group must be served before serving any of the following groups. A group will be happy if they all get fresh donuts. That is, the first customer of the group does not receive a donut that was left over from the previous group. You can freely rearrange the ordering of the groups. Return the maximum possible number of happy groups after rearranging the groups. Example 1: Input: batchSize = 3, groups = [1,2,3,4,5,6] Output: 4 Explanation: You can arrange the groups as [6,2,4,5,1,3]. Then the 1st, 2nd, 4th, and 6th groups will be happy.

Example 2: Input: batchSize = 4, groups = [1,3,2,5,2,2,1,6] Output: 4

Constraints:

1 <= batchSize <= 9 1 <= groups.length <= 30 1 <= groups[i] <= 109

==
Problem Number: 1981 URL: https://leetcode.com/problems/number-of-different-subsequences-gcds Title: 1819. Number of Different Subsequences GCDs Problem Description: You are given an array nums that consists of positive integers. The GCD of a sequence of numbers is defined as the greatest integer that divides all the numbers in the sequence evenly.

For example, the GCD of the sequence [4,6,16] is 2.

A subsequence of an array is a sequence that can be formed by removing some elements (possibly none) of the array.

For example, [2,5,10] is a subsequence of [1,2,1,2,4,1,5,10].

Return the number of different GCDs among all non-empty subsequences of nums. Example 1:

Input: nums = [6,10,3] Output: 5 Explanation: The figure shows all the non-empty subsequences and their GCDs. The different GCDs are 6, 10, 3, 2, and 1.

Example 2: Input: nums = [5,15,40,5,6] Output: 7

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 2 * 105

==
Problem Number: 1982 URL: https://leetcode.com/problems/finding-mk-average Title: 1825. Finding MK Average Problem Description: You are given two integers, m and k, and a stream of integers. You are tasked to implement a data structure that calculates the MKAverage for the stream. The MKAverage can be calculated using these steps:

If the number of the elements in the stream is less than m you should consider the MKAverage to be -1. Otherwise, copy the last m elements of the stream to a separate container. Remove the smallest k elements and the largest k elements from the container. Calculate the average value for the rest of the elements rounded down to the nearest integer.

Implement the MKAverage class:

MKAverage(int m, int k) Initializes the MKAverage object with an empty stream and the two integers m and k. void addElement(int num) Inserts a new element num into the stream. int calculateMKAverage() Calculates and returns the MKAverage for the current stream rounded down to the nearest integer.

Example 1: Input ["MKAverage", "addElement", "addElement", "calculateMKAverage", "addElement", "calculateMKAverage", "addElement", "addElement", "addElement", "calculateMKAverage"] [[3, 1], [3], [1], [], [10], [], [5], [5], [5], []] Output [null, null, null, -1, null, 3, null, null, null, 5]

Explanation MKAverage obj = new MKAverage(3, 1); obj.addElement(3); // current elements are [3] obj.addElement(1); // current elements are [3,1] obj.calculateMKAverage(); // return -1, because m = 3 and only 2 elements exist. obj.addElement(10); // current elements are [3,1,10] obj.calculateMKAverage(); // The last 3 elements are [3,1,10]. // After removing smallest and largest 1 element the container will be [3]. // The average of [3] equals 3/1 = 3, return 3 obj.addElement(5); // current elements are [3,1,10,5] obj.addElement(5); // current elements are [3,1,10,5,5] obj.addElement(5); // current elements are [3,1,10,5,5,5] obj.calculateMKAverage(); // The last 3 elements are [5,5,5]. // After removing smallest and largest 1 element the container will be [5]. // The average of [5] equals 5/1 = 5, return 5

Constraints:

3 <= m <= 105 1 <= k*2 < m 1 <= num <= 105 At most 105 calls will be made to addElement and calculateMKAverage.

==
Problem Number: 1983 URL: https://leetcode.com/problems/minimum-number-of-operations-to-make-string-sorted Title: 1830. Minimum Number of Operations to Make String Sorted Problem Description: You are given a string s (0-indexed). You are asked to perform the following operation on s until you get a sorted string:

Find the largest index i such that 1 <= i < s.length and s[i] < s[i - 1]. Find the largest index j such that i <= j < s.length and s[k] < s[i - 1] for all the possible values of k in the range [i, j] inclusive. Swap the two characters at indices i - 1 and j. Reverse the suffix starting at index i.

Return the number of operations needed to make the string sorted. Since the answer can be too large, return it modulo 109 + 7. Example 1: Input: s = "cba" Output: 5 Explanation: The simulation goes as follows: Operation 1: i=2, j=2. Swap s[1] and s[2] to get s="cab", then reverse the suffix starting at 2. Now, s="cab". Operation 2: i=1, j=2. Swap s[0] and s[2] to get s="bac", then reverse the suffix starting at 1. Now, s="bca". Operation 3: i=2, j=2. Swap s[1] and s[2] to get s="bac", then reverse the suffix starting at 2. Now, s="bac". Operation 4: i=1, j=1. Swap s[0] and s[1] to get s="abc", then reverse the suffix starting at 1. Now, s="acb". Operation 5: i=2, j=2. Swap s[1] and s[2] to get s="abc", then reverse the suffix starting at 2. Now, s="abc".

Example 2: Input: s = "aabaa" Output: 2 Explanation: The simulation goes as follows: Operation 1: i=3, j=4. Swap s[2] and s[4] to get s="aaaab", then reverse the substring starting at 3. Now, s="aaaba". Operation 2: i=4, j=4. Swap s[3] and s[4] to get s="aaaab", then reverse the substring starting at 4. Now, s="aaaab".

Constraints:

1 <= s.length <= 3000 s consists only of lowercase English letters.

==

Problem Number: 1984 URL: https://leetcode.com/problems/find-xor-sum-of-all-pairs-bitwise-and Title: 1835. Find XOR Sum of All Pairs Bitwise AND Problem Description: The XOR sum of a list is the bitwise XOR of all its elements. If the list only contains one element, then its XOR sum will be equal to this element.

For example, the XOR sum of [1,2,3,4] is equal to 1 XOR 2 XOR 3 XOR 4 = 4, and the XOR sum of [3] is equal to 3.

You are given two 0-indexed arrays arr1 and arr2 that consist only of non-negative integers. Consider the list containing the result of arr1[i] AND arr2[j] (bitwise AND) for every (i, j) pair where 0 <= i < arr1.length and 0 <= j < arr2.length. Return the XOR sum of the aforementioned list. Example 1: Input: arr1 = [1,2,3], arr2 = [6,5] Output: 0 Explanation: The list = [1 AND 6, 1 AND 5, 2 AND 6, 2 AND 5, 3 AND 6, 3 AND 5] = [0,1,2,0,2,1]. The XOR sum = 0 XOR 1 XOR 2 XOR 0 XOR 2 XOR 1 = 0.

Example 2: Input: arr1 = [12], arr2 = [4] Output: 4 Explanation: The list = [12 AND 4] = [4]. The XOR sum = 4.

Constraints:

1 <= arr1.length, arr2.length <= 105 0 <= arr1[i], arr2[j] <= 109

==

Problem Number: 1985 URL: https://leetcode.com/problems/maximum-building-height Title: 1840. Maximum Building Height Problem Description: You want to build n new buildings in a city. The new buildings will be built in a line and are labeled from 1 to n. However, there are city restrictions on the heights of the new buildings:

The height of each building must be a non-negative integer. The height of the first building must be 0. The height difference between any two adjacent buildings cannot exceed 1.

Additionally, there are city restrictions on the maximum height of specific buildings. These restrictions are given as a 2D integer array restrictions where restrictions[i] = [idi, maxHeighti] indicates that building idi must have a height less than or equal to maxHeighti. It is guaranteed that each building will appear at most once in restrictions, and building 1 will not be in restrictions. Return the maximum possible height of the tallest building. Example 1:

Input: n = 5, restrictions = [[2,1],[4,1]] Output: 2 Explanation: The green area in the image indicates the maximum allowed height for each building. We can build the buildings with heights [0,1,2,1,2], and the tallest building has a height of 2. Example 2:

Input: n = 6, restrictions = [] Output: 5 Explanation: The green area in the image indicates the maximum allowed height for each building. We can build the buildings with heights [0,1,2,3,4,5], and the tallest building has a height of 5.

Example 3:

Input: n = 10, restrictions = [[5,3],[2,5],[7,4],[10,3]] Output: 5 Explanation: The green area in the image indicates the maximum allowed height for each building. We can build the buildings with heights [0,1,2,3,3,4,4,5,4,3], and the tallest building has a height of 5.

Constraints:

2 <= n <= 109 0 <= restrictions.length <= min(n - 1, 105) 2 <= idi <= n idi is unique. 0 <= maxHeighti <= 109

==
Problem Number: 1986 URL: https://leetcode.com/problems/closest-room Title: 1847. Closest Room Problem Description: There is a hotel with n rooms. The rooms are represented by a 2D integer array rooms where rooms[i] = [roomIdi, sizei] denotes that there is a room with room number roomIdi and size equal to sizei. Each roomIdi is guaranteed to be unique. You are also given k queries in a 2D array queries where queries[j] = [preferredj, minSizej]. The answer to the jth query is the room number id of a room such that:

The room has a size of at least minSizej, and abs(id - preferredj) is minimized, where abs(x) is the absolute value of x.

If there is a tie in the absolute difference, then use the room with the smallest such id. If there is no such room, the answer is -1. Return an array answer of length k where answer[j] contains the answer to the jth query. Example 1: Input: rooms = [[2,2],[1,2],[3,2]], queries = [[3,1],[3,3],[5,2]] Output: [3,-1,3] Explanation: The answers to the queries are as follows: Query = [3,1]: Room number 3 is the closest as abs(3 - 3) = 0, and its size of 2 is at least 1. The answer is 3. Query = [3,3]: There are no rooms with a size of at least 3, so the answer is -1. Query = [5,2]: Room number 3 is the closest as abs(3 - 5) = 2, and its size of 2 is at least 2. The answer is 3. Example 2: Input: rooms = [[1,4],[2,3],[3,5],[4,1],[5,2]], queries = [[2,3],[2,4],[2,5]] Output: [2,1,3] Explanation: The answers to the queries are as follows: Query = [2,3]: Room number 2 is the closest as abs(2 - 2) = 0, and its size of 3 is at least 3. The answer is 2. Query = [2,4]: Room numbers 1 and 3 both have sizes of at least 4. The answer is 1 since it is smaller. Query = [2,5]: Room number 3 is the only room with a size of at least 5. The answer is 3. Constraints:

n == rooms.length 1 <= n <= 105 k == queries.length 1 <= k <= 104 1 <= roomIdi, preferredj <= 107 1 <= sizei, minSizej <= 107

==
Problem Number: 1987 URL: https://leetcode.com/problems/minimum-interval-to-include-each-query Title: 1851. Minimum Interval to Include Each Query Problem Description: You are given a 2D integer array intervals, where

intervals[i] = [lefti, righti] describes the ith interval starting at lefti and ending at righti (inclusive). The size of an interval is defined as the number of integers it contains, or more formally righti - lefti + 1. You are also given an integer array queries. The answer to the jth query is the size of the smallest interval i such that lefti <= queries[j] <= righti. If no such interval exists, the answer is -1. Return an array containing the answers to the queries. Example 1: Input: intervals = [[1,4],[2,4],[3,6],[4,4]], queries = [2,3,4,5] Output: [3,3,1,4] Explanation: The queries are processed as follows: - Query = 2: The interval [2,4] is the smallest interval containing 2. The answer is 4 - 2 + 1 = 3. - Query = 3: The interval [2,4] is the smallest interval containing 3. The answer is 4 - 2 + 1 = 3. - Query = 4: The interval [4,4] is the smallest interval containing 4. The answer is 4 - 4 + 1 = 1. - Query = 5: The interval [3,6] is the smallest interval containing 5. The answer is 6 - 3 + 1 = 4.

Example 2: Input: intervals = [[2,3],[2,5],[1,8],[20,25]], queries = [2,19,5,22] Output: [2,-1,4,6] Explanation: The queries are processed as follows: - Query = 2: The interval [2,3] is the smallest interval containing 2. The answer is 3 - 2 + 1 = 2. - Query = 19: None of the intervals contain 19. The answer is -1. - Query = 5: The interval [2,5] is the smallest interval containing 5. The answer is 5 - 2 + 1 = 4. - Query = 22: The interval [20,25] is the smallest interval containing 22. The answer is 25 - 20 + 1 = 6.

Constraints:

1 <= intervals.length <= 105 1 <= queries.length <= 105 intervals[i].length == 2 1 <= lefti <= righti <= 107 1 <= queries[j] <= 107

==
Problem Number: 1988 URL: https://leetcode.com/problems/largest-color-value-in-a-directed-graph Title: 1857. Largest Color Value in a Directed Graph Problem Description: There is a directed graph of n colored nodes and m edges. The nodes are numbered from 0 to n - 1. You are given a string colors where colors[i] is a lowercase English letter representing the color of the ith node in this graph (0-indexed). You are also given a 2D array edges where edges[j] = [aj, bj] indicates that there is a directed edge from node aj to node bj. A valid path in the graph is a sequence of nodes x1 -> x2 -> x3 -> ... -> xk such that there is a directed edge from xi to xi+1 for every 1 <= i < k. The color value of the path is the number of nodes that are colored the most frequently occurring color along that path. Return the largest color value of any valid path in the given graph, or -1 if the graph contains a cycle. Example 1:

Input: colors = "abaca", edges = [[0,1],[0,2],[2,3],[3,4]] Output: 3 Explanation: The path 0 -> 2 -> 3 -> 4 contains 3 nodes that are colored "a" (red in the above image).

Example 2:

Input: colors = "a", edges = [[0,0]] Output: -1 Explanation: There is a cycle from 0 to 0.

Constraints:

n == colors.length m == edges.length 1 <= n <= 105 0 <= m <= 105 colors consists of lowercase English letters. 0 <= aj, bj < n

==
Problem Number: 1989 URL: https://leetcode.com/problems/sum-of-floored-pairs Title: 1862. Sum of Floored Pairs Problem Description: Given an integer array nums, return the sum of floor(nums[i] / nums[j]) for all pairs of indices 0 <= i, j < nums.length in the array. Since the answer may be too large, return it modulo 109 + 7. The floor() function returns the integer part of the division. Example 1: Input: nums = [2,5,9] Output: 10 Explanation: floor(2 / 5) = floor(2 / 9) = floor(5 / 9) = 0 floor(2 / 2) = floor(5 / 5) = floor(9 / 9) = 1 floor(5 / 2) = 2 floor(9 / 2) = 4 floor(9 / 5) = 1 We calculate the floor of the division for every pair of indices in the array then sum them up.

Example 2: Input: nums = [7,7,7,7,7,7,7] Output: 49

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105

==
Problem Number: 1990 URL: https://leetcode.com/problems/number-of-ways-to-rearrange-sticks-with-k-sticks-visible Title: 1866. Number of Ways to Rearrange Sticks With K Sticks Visible Problem Description: There are n uniquely-sized sticks whose lengths are integers from 1 to n. You want to arrange the sticks such that exactly k sticks are visible from the left. A stick is visible from the left if there are no longer sticks to the left of it.

For example, if the sticks are arranged [1,3,2,5,4], then the sticks with lengths 1, 3, and 5 are visible from the left.

Given n and k, return the number of such arrangements. Since the answer may be large, return it modulo 109 + 7. Example 1: Input: n = 3, k = 2 Output: 3 Explanation: [1,3,2], [2,3,1], and [2,1,3] are the only arrangements such that exactly 2 sticks are visible. The visible sticks are underlined.

Example 2: Input: n = 5, k = 5 Output: 1 Explanation: [1,2,3,4,5] is the only arrangement such that all 5 sticks are visible. The visible sticks are underlined.

Example 3: Input: n = 20, k = 11 Output: 647427950 Explanation: There are 647427950 (mod 109 + 7) ways to rearrange the sticks such that exactly 11 sticks are visible.

Constraints:

1 <= n <= 1000 1 <= k <= n

==
Problem Number: 1991 URL: https://leetcode.com/problems/stone-game-viii Title: 1872. Stone Game VIII Problem Description: Alice and Bob take turns

playing a game, with Alice starting first. There are n stones arranged in a row. On each player's turn, while the number of stones is more than one, they will do the following:

Choose an integer x > 1, and remove the leftmost x stones from the row. Add the sum of the removed stones' values to the player's score. Place a new stone, whose value is equal to that sum, on the left side of the row.

The game stops when only one stone is left in the row. The score difference between Alice and Bob is (Alice's score - Bob's score). Alice's goal is to maximize the score difference, and Bob's goal is the minimize the score difference. Given an integer array stones of length n where stones[i] represents the value of the ith stone from the left, return the score difference between Alice and Bob if they both play optimally. Example 1: Input: stones = [-1,2,-3,4,-5] Output: 5 Explanation: - Alice removes the first 4 stones, adds (-1) + 2 + (-3) + 4 = 2 to her score, and places a stone of value 2 on the left. stones = [2,-5]. - Bob removes the first 2 stones, adds 2 + (-5) = -3 to his score, and places a stone of value -3 on the left. stones = [-3]. The difference between their scores is 2 - (-3) = 5.

Example 2: Input: stones = [7,-6,5,10,5,-2,-6] Output: 13 Explanation: - Alice removes all stones, adds 7 + (-6) + 5 + 10 + 5 + (-2) + (-6) = 13 to her score, and places a stone of value 13 on the left. stones = [13]. The difference between their scores is 13 - 0 = 13.

Example 3: Input: stones = [-10,-12] Output: -22 Explanation: - Alice can only make one move, which is to remove both stones. She adds (-10) + (-12) = -22 to her score and places a stone of value -22 on the left. stones = [-22]. The difference between their scores is (-22) - 0 = -22.

Constraints:

n == stones.length 2 <= n <= 105 -104 <= stones[i] <= 104

==

Problem Number: 1992 URL: https://leetcode.com/problems/minimum-xor-sum-of-two-arrays Title: 1879. Minimum XOR Sum of Two Arrays Problem Description: You are given two integer arrays nums1 and nums2 of length n. The XOR sum of the two integer arrays is (nums1[0] XOR nums2[0]) + (nums1[1] XOR nums2[1]) + ... + (nums1[n - 1] XOR nums2[n - 1]) (0-indexed).

For example, the XOR sum of [1,2,3] and [3,2,1] is equal to (1 XOR 3) + (2 XOR 2) + (3 XOR 1) = 2 + 0 + 2 = 4.

Rearrange the elements of nums2 such that the resulting XOR sum is minimized. Return the XOR sum after the rearrangement. Example 1: Input: nums1 = [1,2], nums2 = [2,3] Output: 2 Explanation: Rearrange nums2 so that it becomes [3,2]. The XOR sum is (1 XOR 3) + (2 XOR 2) = 2 + 0 = 2. Example 2: Input: nums1 = [1,0,3], nums2 = [5,3,4] Output: 8 Explanation: Rearrange nums2 so that it becomes [5,4,3]. The XOR sum is (1 XOR 5) + (0 XOR 4) + (3 XOR 3) = 4 + 4 + 0 = 8.

Constraints:

n == nums1.length n == nums2.length 1 <= n <= 14 0 <= nums1[i], nums2[i] <= 107

==

Problem Number: 1993 URL: https://leetcode.com/problems/minimum-skips-to-arrive-at-meeting-on-time Title: 1883. Minimum Skips to Arrive at Meeting On Time Problem Description: You are given an integer hoursBefore, the number of hours you have to travel to your meeting. To arrive at your meeting, you have to travel through n roads. The road lengths are given as an integer array dist of length n, where dist[i] describes the length of the ith road in kilometers. In addition, you are given an integer speed, which is the speed (in km/h) you will travel at. After you travel road i, you must rest and wait for the next integer hour before you can begin traveling on the next road. Note that you do not have to rest after traveling the last road because you are already at the meeting.

For example, if traveling a road takes 1.4 hours, you must wait until the 2 hour mark before traveling the next road. If traveling a road takes exactly 2 hours, you do not need to wait.

However, you are allowed to skip some rests to be able to arrive on time, meaning you do not need to wait for the next integer hour. Note that this means you may finish traveling future roads at different hour marks.

For example, suppose traveling the first road takes 1.4 hours and traveling the second road takes 0.6 hours. Skipping the rest after the first road will mean you finish traveling the second road right at the 2 hour mark, letting you start traveling the third road immediately.

Return the minimum number of skips required to arrive at the meeting on time, or -1 if it is impossible. Example 1: Input: dist = [1,3,2], speed = 4, hoursBefore = 2 Output: 1 Explanation: Without skipping any rests, you will arrive in (1/4 + 3/4) + (3/4 + 1/4) + (2/4) = 2.5 hours. You can skip the first rest to arrive in ((1/4 + 0) + (3/4 + 0)) + (2/4) = 1.5 hours. Note that the second rest is shortened because you finish traveling the second road at an integer hour due to skipping the first rest.

Example 2: Input: dist = [7,3,5,5], speed = 2, hoursBefore = 10 Output: 2 Explanation: Without skipping any rests, you will arrive in (7/2 + 1/2) + (3/2 + 1/2) + (5/2 + 1/2) + (5/2) = 11.5 hours. You can skip the first and third rest to arrive in ((7/2 + 0) + (3/2 + 0)) + ((5/2 + 0) + (5/2)) = 10 hours.

Example 3: Input: dist = [7,3,5,5], speed = 1, hoursBefore = 10 Output: -1 Explanation: It is impossible to arrive at the meeting on time even if you skip all the rests.

Constraints:

n == dist.length 1 <= n <= 1000 1 <= dist[i] <= 105 1 <= speed <= 106 1 <= hoursBefore <= 107

== Problem Number: 1994 URL: https://leetcode.com/problems/minimum-space-wasted-from-packaging Title: 1889. Minimum Space Wasted From Packaging Problem Description: You have n packages that you are trying to place in boxes, one package in each box. There are m suppliers that each produce boxes of different sizes (with infinite supply). A package can be placed in a box if the size of the package is less than or equal to the size of the box. The package sizes are given as an integer array packages, where packages[i] is the size of the ith package. The suppliers are given as a 2D integer array boxes, where boxes[j] is an array of box sizes that the jth supplier produces. You want to choose a single supplier and use boxes from them such that the total wasted space is minimized. For each package in a box, we define the space wasted to be size of the box - size of the package. The total wasted space is the sum of the space wasted in all the boxes.

For example, if you have to fit packages with sizes [2,3,5] and the supplier offers boxes of sizes [4,8], you can fit the packages of size-2 and size-3 into two boxes of size-4 and the package with size-5 into a box of size-8. This would result in a waste of (4-2) + (4-3) + (8-5) = 6.

Return the minimum total wasted space by choosing the box supplier optimally, or -1 if it is impossible to fit all the packages inside boxes. Since the answer may be large, return it modulo 109 + 7. Example 1: Input: packages = [2,3,5], boxes = [[4,8],[2,8]] Output: 6 Explanation: It is optimal to choose the first supplier, using two size-4 boxes and one size-8 box. The total waste is (4-2) + (4-3) + (8-5) = 6.

Example 2: Input: packages = [2,3,5], boxes = [[1,4],[2,3],[3,4]] Output: -1 Explanation: There is no box that the package of size 5 can fit in.

Example 3: Input: packages = [3,5,8,10,11,12], boxes = [[12],[11,9],[10,5,14]] Output: 9 Explanation: It is optimal to choose the third supplier, using two size-5 boxes, two size-10 boxes, and two size-14 boxes. The total waste is (5-3) + (5-5) + (10-8) + (10-10) + (14-11) + (14-12) = 9.

Constraints:

n == packages.length m == boxes.length 1 <= n <= 105 1 <= m <= 105 1 <= packages[i] <= 105 1 <= boxes[j].length <= 105 1 <= boxes[j][k] <= 105 sum(boxes[j].length) <= 105 The elements in boxes[j] are distinct.

== Problem Number: 1995 URL: https://leetcode.com/problems/minimum-cost-to-change-the-final-value-of-expression Title: 1896. Minimum Cost to Change the Final Value of Expression Problem Description: You are given a valid boolean expression as a string expression consisting of the characters '1','0','&' (bitwise AND operator),'|' (bitwise OR operator),'(', and ')'.

For example, "()1|1" and "(1)&()" are not valid while "1", "(((1))|(0))", and "1|(0&(1))" are valid expressions.

Return the minimum cost to change the final value of the expression.

For example, if expression = "1|1|(0&0)&1", its value is 1|1|(0&0)&1 = 1|1|0&1 = 1|0&1 = 1&1 = 1. We want to apply operations so that the new expression evaluates to 0.

The cost of changing the final value of an expression is the number of operations performed on the expression. The types of operations are described as follows:

Turn a '1' into a '0'. Turn a '0' into a '1'. Turn a '&' into a '|'. Turn a '|' into a '&'.

Note: '&' does not take precedence over '|' in the order of calculation. Evaluate parentheses first, then in left-to-right order.   Example 1: Input: expression = "1&(0|1)" Output: 1 Explanation: We can turn "1&(0|1)" into "1&(0&1)" by changing the '|' to a '&' using 1 operation. The new expression evaluates to 0.

Example 2: Input: expression = "(0&0)&(0&0&0)" Output: 3 Explanation: We can turn "(0&0)&(0&0&0)" into "(0|1)|(0&0&0)" using 3 operations. The new expression evaluates to 1.

Example 3: Input: expression = "(0|(1|0&1))" Output: 1 Explanation: We can turn "(0|(1|0&1))" into "(0|(0|0&1))" using 1 operation. The new expression evaluates to 0.   Constraints:

1 <= expression.length <= 105 expression only contains '1','0','&','|','(', and ')' All parentheses are properly matched. There will be no empty parentheses (i.e: "()" is not a substring of expression).

==
Problem Number: 1996 URL: https://leetcode.com/problems/the-earliest-and-latest-rounds-where-players-compete Title: 1900. The Earliest and Latest Rounds Where Players Compete Problem Description: There is a tournament where n players are participating. The players are standing in a single row and are numbered from 1 to n based on their initial standing position (player 1 is the first player in the row, player 2 is the second player in the row, etc.). The tournament consists of multiple rounds (starting from round number 1). In each round, the ith player from the front of the row competes against the ith player from the end of the row, and the winner advances to the next round. When the number of players is odd for the current round, the player in the middle automatically advances to the next round.

For example, if the row consists of players 1, 2, 4, 6, 7

Player 1 competes against player 7. Player 2 competes against player 6. Player 4 automatically advances to the next round.

After each round is over, the winners are lined back up in the row based on the original ordering assigned to them initially (ascending order). The players numbered firstPlayer and secondPlayer are the best in the tournament. They can win against any other player before they compete against each other. If any two other players compete against each other, either of them might win, and thus you may choose the outcome of this round. Given the integers n, firstPlayer, and secondPlayer, return an integer array containing two values, the earliest possible round number and the latest possible round number in which these two players will compete against each other, respectively. Example 1: Input: n = 11, firstPlayer = 2, secondPlayer = 4 Output: [3,4] Explanation: One possible scenario which leads to the earliest round number: First round: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 Second round: 2, 3, 4, 5, 6, 11 Third round: 2, 3, 4 One possible scenario which leads to the latest round number: First round: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 Second round: 1, 2, 3, 4, 5, 6 Third round: 1, 2, 4 Fourth round: 2, 4

Example 2: Input: n = 5, firstPlayer = 1, secondPlayer = 5 Output: [1,1] Explanation: The players numbered 1 and 5 compete in the first round. There is no way to make them compete in any other round.

Constraints:

2 <= n <= 28 1 <= firstPlayer < secondPlayer <= n

==

Problem Number: 1997 URL: https://leetcode.com/problems/design-movie-rental-system Title: 1912. Design Movie Rental System Problem Description: You have a movie renting company consisting of n shops. You want to implement a renting system that supports searching for, booking, and returning movies. The system should also support generating a report of the currently rented movies. Each movie is given as a 2D integer array entries where entries[i] = [shopi, moviei, pricei] indicates that there is a copy of movie moviei at shop shopi with a rental price of pricei. Each shop carries at most one copy of a movie moviei. The system should support the following functions:

Search: Finds the cheapest 5 shops that have an unrented copy of a given movie. The shops should be sorted by price in ascending order, and in case of a tie, the one with the smaller shopi should appear first. If there are less than 5 matching shops, then all of them should be returned. If no shop has an unrented copy, then an empty list should be returned. Rent: Rents an unrented copy of a given movie from a given shop. Drop: Drops off a previously rented copy of a given movie at a given shop. Report: Returns the cheapest 5 rented movies (possibly of the same movie ID) as a 2D list res where res[j] = [shopj, moviej] describes that the jth cheapest rented movie moviej was rented from the shop shopj. The movies in res should be sorted by price in ascending order, and in case of a tie, the one with the smaller shopj should appear first, and if there is still tie, the one with the smaller moviej should appear first. If there are fewer than 5 rented movies, then all of them should be returned. If no movies are currently being rented, then an empty list should be returned.

Implement the MovieRentingSystem class:

MovieRentingSystem(int n, int[][] entries) Initializes the MovieRentingSystem object with n shops and the movies in entries. List<Integer> search(int movie) Returns a list of shops that have an unrented copy of the given movie as described above. void rent(int shop, int movie) Rents the given movie from the given shop. void drop(int shop, int movie) Drops off a previously rented movie at the given shop. List<List<Integer» report() Returns a list of cheapest rented movies as described above.

Note: The test cases will be generated such that rent will only be called if the shop has an unrented copy of the movie, and drop will only be called if the shop had previously rented out the movie. Example 1: Input ["MovieRentingSystem", "search", "rent", "rent", "report", "drop", "search"] [[3, [[0, 1, 5], [0, 2, 6], [0, 3, 7], [1, 1, 4], [1, 2, 7], [2, 1, 5]]], [1], [0, 1], [1, 2], [], [1, 2], [2]] Output [null, [1, 0, 2], null, null, [[0, 1], [1, 2]], null, [0, 1]]

Explanation MovieRentingSystem movieRentingSystem = new MovieRentingSystem(3, [[0, 1, 5], [0, 2, 6], [0, 3, 7], [1, 1, 4], [1, 2, 7], [2, 1, 5]]); movieRentingSystem.search(1); // return [1, 0, 2], Movies of ID 1 are unrented at shops 1, 0, and 2. Shop 1 is cheapest; shop 0 and 2 are the same price, so order by shop number. movieRentingSystem.rent(0, 1); // Rent movie 1 from shop 0. Unrented movies at shop 0 are now [2,3]. movieRentingSystem.rent(1, 2); // Rent movie 2 from shop 1. Unrented movies at shop 1 are now [1]. movieRentingSystem.report(); // return [[0, 1], [1, 2]]. Movie 1 from shop 0 is cheapest, followed by movie 2 from shop 1. movieRentingSystem.drop(1, 2); // Drop off movie 2 at shop 1. Unrented movies at shop 1 are now [1,2]. movieRentingSystem.search(2); // return [0, 1]. Movies of ID 2 are unrented at shops 0 and 1. Shop 0 is cheapest, followed by shop 1.

Constraints:

1 <= n <= 3 * 105 1 <= entries.length <= 105 0 <= shopi < n 1 <= moviei, pricei <= 104 Each shop carries at most one copy of a movie moviei. At most 105 calls in total will be made to search, rent, drop and report.

==
Problem Number: 1998 URL: https://leetcode.com/problems/count-ways-to-build-rooms-in-an-ant-colony Title: 1916. Count Ways to Build Rooms in an Ant Colony Problem Description: You are an ant tasked with adding n new rooms numbered 0 to n-1 to your colony. You are given the expansion plan as a 0-indexed integer array of length n, prevRoom, where prevRoom[i] indicates that you must build room prevRoom[i] before building room i, and these two rooms must be connected directly. Room 0 is already built, so prevRoom[0] = -1. The expansion plan is given such that once all the rooms are built, every room will be reachable from room 0. You can only build one room at a time, and you can travel freely between rooms you have already built only if they are

connected. You can choose to build any room as long as its previous room is already built. Return the number of different orders you can build all the rooms in. Since the answer may be large, return it modulo 109 + 7. Example 1:

Input: prevRoom = [-1,0,1] Output: 1 Explanation: There is only one way to build the additional rooms: 0 → 1 → 2

Example 2:

Input: prevRoom = [-1,0,0,1,2] Output: 6 Explanation: The 6 ways are: 0 → 1 → 3 → 2 → 4 0 → 2 → 4 → 1 → 3 0 → 1 → 2 → 3 → 4 0 → 1 → 2 → 4 → 3 0 → 2 → 1 → 3 → 4 0 → 2 → 1 → 4 → 3

Constraints:

n == prevRoom.length 2 <= n <= 105 prevRoom[0] == -1 0 <= prevRoom[i] < n for all 1 <= i < n Every room is reachable from room 0 once all the rooms are built.

==
Problem Number: 1999 URL: https://leetcode.com/problems/longest-common-subpath Title: 1923. Longest Common Subpath Problem Description: There is a country of n cities numbered from 0 to n - 1. In this country, there is a road connecting every pair of cities. There are m friends numbered from 0 to m - 1 who are traveling through the country. Each one of them will take a path consisting of some cities. Each path is represented by an integer array that contains the visited cities in order. The path may contain a city more than once, but the same city will not be listed consecutively. Given an integer n and a 2D integer array paths where paths[i] is an integer array representing the path of the ith friend, return the length of the longest common subpath that is shared by every friend's path, or 0 if there is no common subpath at all. A subpath of a path is a contiguous sequence of cities within that path. Example 1: Input: n = 5, paths = [[0,1,2,3,4], [2,3,4], [4,0,1,2,3]] Output: 2 Explanation: The longest common subpath is [2,3].

Example 2: Input: n = 3, paths = [[0],[1],[2]] Output: 0 Explanation: There is no common subpath shared by the three paths.

Example 3: Input: n = 5, paths = [[0,1,2,3,4], [4,3,2,1,0]] Output: 1 Explanation: The possible longest common subpaths are [0], [1], [2], [3], and [4]. All have a length of 1. Constraints:

1 <= n <= 105 m == paths.length 2 <= m <= 105 sum(paths[i].length) <= 105 0 <= paths[i][j] < n The same city is not listed multiple times consecutively in paths[i].

==
Problem Number: 2000 URL: https://leetcode.com/problems/minimum-cost-to-reach-destination-in-time Title: 1928. Minimum Cost to Reach Destination in Time Problem Description: There is a country of n cities numbered from 0

to n - 1 where all the cities are connected by bi-directional roads. The roads are represented as a 2D integer array edges where edges[i] = [xi, yi, timei] denotes a road between cities xi and yi that takes timei minutes to travel. There may be multiple roads of differing travel times connecting the same two cities, but no road connects a city to itself. Each time you pass through a city, you must pay a passing fee. This is represented as a 0-indexed integer array passingFees of length n where passingFees[j] is the amount of dollars you must pay when you pass through city j. In the beginning, you are at city 0 and want to reach city n - 1 in maxTime minutes or less. The cost of your journey is the summation of passing fees for each city that you passed through at some moment of your journey (including the source and destination cities). Given maxTime, edges, and passingFees, return the minimum cost to complete your journey, or -1 if you cannot complete it within maxTime minutes.   Example 1:

Input: maxTime = 30, edges = [[0,1,10],[1,2,10],[2,5,10],[0,3,1],[3,4,10],[4,5,15]], passingFees = [5,1,2,20,20,3] Output: 11 Explanation: The path to take is 0 -> 1 -> 2 -> 5, which takes 30 minutes and has $11 worth of passing fees.

Example 2:

Input: maxTime = 29, edges = [[0,1,10],[1,2,10],[2,5,10],[0,3,1],[3,4,10],[4,5,15]], passingFees = [5,1,2,20,20,3] Output: 48 Explanation: The path to take is 0 -> 3 -> 4 -> 5, which takes 26 minutes and has $48 worth of passing fees. You cannot take path 0 -> 1 -> 2 -> 5 since it would take too long.

Example 3: Input: maxTime = 25, edges = [[0,1,10],[1,2,10],[2,5,10],[0,3,1],[3,4,10],[4,5,15]], passingFees = [5,1,2,20,20,3] Output: -1 Explanation: There is no way to reach city 5 from city 0 within 25 minutes.

Constraints:

1 <= maxTime <= 1000 n == passingFees.length 2 <= n <= 1000 n - 1 <= edges.length <= 1000 0 <= xi, yi <= n - 1 1 <= timei <= 1000 1 <= passingFees[j] <= 1000 The graph may contain multiple edges between two nodes. The graph does not contain self loops.

==
Problem Number: 2001 URL: https://leetcode.com/problems/painting-a-grid-with-three-different-colors Title: 1931. Painting a Grid With Three Different Colors Problem Description: You are given two integers m and n. Consider an m x n grid where each cell is initially white. You can paint each cell red, green, or blue. All cells must be painted. Return the number of ways to color the grid with no two adjacent cells having the same color. Since the answer can be very large, return it modulo 109 + 7.   Example 1:

Input: m = 1, n = 1 Output: 3 Explanation: The three possible colorings are shown in the image above.

Example 2:

Input: m = 1, n = 2 Output: 6 Explanation: The six possible colorings are shown in the image above.

Example 3: Input: m = 5, n = 5 Output: 580986

Constraints:

1 <= m <= 5 1 <= n <= 1000

==
Problem Number: 2002 URL: https://leetcode.com/problems/merge-bsts-to-create-single-bst Title: 1932. Merge BSTs to Create Single BST Problem Description: You are given n BST (binary search tree) root nodes for n separate BSTs stored in an array trees (0-indexed). Each BST in trees has at most 3 nodes, and no two roots have the same value. In one operation, you can:

Select two distinct indices i and j such that the value stored at one of the leaves of trees[i] is equal to the root value of trees[j]. Replace the leaf node in trees[i] with trees[j]. Remove trees[j] from trees.

Return the root of the resulting BST if it is possible to form a valid BST after performing n - 1 operations, or null if it is impossible to create a valid BST. A BST (binary search tree) is a binary tree where each node satisfies the following property:

Every node in the node's left subtree has a value strictly less than the node's value. Every node in the node's right subtree has a value strictly greater than the node's value.

A leaf is a node that has no children.   Example 1:

Input: trees = [[2,1],[3,2,5],[5,4]] Output: [3,2,5,1,null,4] Explanation: In the first operation, pick i=1 and j=0, and merge trees[0] into trees[1]. Delete trees[0], so trees = [[3,2,5,1],[5,4]].

In the second operation, pick i=0 and j=1, and merge trees[1] into trees[0]. Delete trees[1], so trees = [[3,2,5,1,null,4]].

The resulting tree, shown above, is a valid BST, so return its root. Example 2:

Input: trees = [[5,3,8],[3,2,6]] Output: [] Explanation: Pick i=0 and j=1 and merge trees[1] into trees[0]. Delete trees[1], so trees = [[5,3,8,2,6]].

The resulting tree is shown above. This is the only valid operation that can be performed, but the resulting tree is not a valid BST, so return null.

Example 3:

Input: trees = [[5,4],[3]] Output: [] Explanation: It is impossible to perform any operations.

Constraints:

n == trees.length 1 <= n <= 5 * 104 The number of nodes in each tree is in the range [1, 3]. Each node in the input may have children but no grandchildren. No two roots of trees have the same value. All the trees in the input are valid BSTs. 1 <= TreeNode.val <= 5 * 104.

==
Problem Number: 2003 URL: https://leetcode.com/problems/maximum-genetic-difference-query Title: 1938. Maximum Genetic Difference Query Problem Description: There is a rooted tree consisting of n nodes numbered 0 to n - 1. Each node's number denotes its unique genetic value (i.e. the genetic value of node x is x). The genetic difference between two genetic values is defined as the bitwise-XOR of their values. You are given the integer array parents, where parents[i] is the parent for node i. If node x is the root of the tree, then parents[x] == -1. You are also given the array queries where queries[i] = [nodei, vali]. For each query i, find the maximum genetic difference between vali and pi, where pi is the genetic value of any node that is on the path between nodei and the root (including nodei and the root). More formally, you want to maximize vali XOR pi. Return an array ans where ans[i] is the answer to the ith query. Example 1:

Input: parents = [-1,0,1,1], queries = [[0,2],[3,2],[2,5]] Output: [2,3,7] Explanation: The queries are processed as follows: - [0,2]: The node with the maximum genetic difference is 0, with a difference of 2 XOR 0 = 2. - [3,2]: The node with the maximum genetic difference is 1, with a difference of 2 XOR 1 = 3. - [2,5]: The node with the maximum genetic difference is 2, with a difference of 5 XOR 2 = 7.

Example 2:

Input: parents = [3,7,-1,2,0,7,0,2], queries = [[4,6],[1,15],[0,5]] Output: [6,14,7] Explanation: The queries are processed as follows: - [4,6]: The node with the maximum genetic difference is 0, with a difference of 6 XOR 0 = 6. - [1,15]: The node with the maximum genetic difference is 1, with a difference of 15 XOR 1 = 14. - [0,5]: The node with the maximum genetic difference is 2, with a difference of 5 XOR 2 = 7.

Constraints:

2 <= parents.length <= 105 0 <= parents[i] <= parents.length - 1 for every node i that is not the root. parents[root] == -1 1 <= queries.length <= 3 * 104 0 <= nodei <= parents.length - 1 0 <= vali <= 2 * 105

==
Problem Number: 2004 URL: https://leetcode.com/problems/number-of-visible-people-in-a-queue Title: 1944. Number of Visible People in a Queue Problem Description: There are n people standing in a queue, and they numbered from 0 to n - 1 in left to right order. You are given an array heights of distinct integers where heights[i] represents the height of the ith person. A person can see another person to their right in the queue if everybody in

between is shorter than both of them. More formally, the ith person can see the jth person if i < j and min(heights[i], heights[j]) > max(heights[i+1], heights[i+2], ..., heights[j-1]). Return an array answer of length n where answer[i] is the number of people the ith person can see to their right in the queue. Example 1:

Input: heights = [10,6,8,5,11,9] Output: [3,1,2,1,1,0] Explanation: Person 0 can see person 1, 2, and 4. Person 1 can see person 2. Person 2 can see person 3 and 4. Person 3 can see person 4. Person 4 can see person 5. Person 5 can see no one since nobody is to the right of them.

Example 2: Input: heights = [5,1,2,3,10] Output: [4,1,1,1,0]

Constraints:

n == heights.length 1 <= n <= 105 1 <= heights[i] <= 105 All the values of heights are unique.

==

Problem Number: 2005 URL: https://leetcode.com/problems/delete-duplicate-folders-in-system Title: 1948. Delete Duplicate Folders in System Problem Description: Due to a bug, there are many duplicate folders in a file system. You are given a 2D array paths, where paths[i] is an array representing an absolute path to the ith folder in the file system.

For example, ["one", "two", "three"] represents the path "/one/two/three".

Two folders (not necessarily on the same level) are identical if they contain the same non-empty set of identical subfolders and underlying subfolder structure. The folders do not need to be at the root level to be identical. If two or more folders are identical, then mark the folders as well as all their subfolders.

For example, folders "/a" and "/b" in the file structure below are identical. They (as well as their subfolders) should all be marked:

/a /a/x /a/x/y /a/z /b /b/x /b/x/y /b/z

However, if the file structure also included the path "/b/w", then the folders "/a" and "/b" would not be identical. Note that "/a/x" and "/b/x" would still be considered identical even with the added folder.

Once all the identical folders and their subfolders have been marked, the file system will delete all of them. The file system only runs the deletion once, so any folders that become identical after the initial deletion are not deleted. Return the 2D array ans containing the paths of the remaining folders after deleting all the marked folders. The paths may be returned in any order. Example 1:

Input: paths = [["a"],["c"],["d"],["a","b"],["c","b"],["d","a"]] Output: [["d"],["d","a"]] Explanation: The file structure is as shown. Folders "/a" and "/c" (and their subfolders) are marked for deletion because they both contain an empty folder named "b".

Example 2:

Input: paths = [["a"],["c"],["a","b"],["c","b"],["a","b","x"],["a","b","x","y"],["w"],["w","y"]] Output: [["c"],["c","b"],["a"],["a","b"]] Explanation: The file structure is as shown. Folders "/a/b/x" and "/w" (and their subfolders) are marked for deletion because they both contain an empty folder named "y". Note that folders "/a" and "/c" are identical after the deletion, but they are not deleted because they were not marked beforehand.

Example 3:

Input: paths = [["a","b"],["c","d"],["c"],["a"]] Output: [["c"],["c","d"],["a"],["a","b"]] Explanation: All folders are unique in the file system. Note that the returned array can be in a different order as the order does not matter.

Constraints:

1 <= paths.length <= 2 * 104 1 <= paths[i].length <= 500 1 <= paths[i][j].length <= 10 1 <= sum(paths[i][j].length) <= 2 * 105 path[i][j] consists of lowercase English letters. No two paths lead to the same folder. For any folder not at the root level, its parent folder will also be in the input.

==

Problem Number: 2006 URL: https://leetcode.com/problems/count-number-of-special-subsequences Title: 1955. Count Number of Special Subsequences Problem Description: A sequence is special if it consists of a positive number of 0s, followed by a positive number of 1s, then a positive number of 2s.

For example, [0,1,2] and [0,0,1,1,1,2] are special. In contrast, [2,1,0], [1], and [0,1,2,0] are not special.

Given an array nums (consisting of only integers 0, 1, and 2), return the number of different subsequences that are special. Since the answer may be very large, return it modulo 109 + 7. A subsequence of an array is a sequence that can be derived from the array by deleting some or no elements without changing the order of the remaining elements. Two subsequences are different if the set of indices chosen are different. Example 1: Input: nums = [0,1,2,2] Output: 3 Explanation: The special subsequences are bolded [0,1,2,2], [0,1,2,2], and [0,1,2,2].

Example 2: Input: nums = [2,2,0,0] Output: 0 Explanation: There are no special subsequences in [2,2,0,0].

Example 3: Input: nums = [0,1,2,0,1,2] Output: 7 Explanation: The special subsequences are bolded: - [0,1,2,0,1,2] - [0,1,2,0,1,2] - [0,1,2,0,1,2] - [0,1,2,0,1,2] - [0,1,2,0,1,2] - [0,1,2,0,1,2] - [0,1,2,0,1,2]

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 2

==
Problem Number: 2007 URL: https://leetcode.com/problems/maximum-product-of-the-length-of-two-palindromic-substrings Title: 1960. Maximum Product of the Length of Two Palindromic Substrings Problem Description: You are given a 0-indexed string s and are tasked with finding two non-intersecting palindromic substrings of odd length such that the product of their lengths is maximized. More formally, you want to choose four integers i, j, k, l such that 0 <= i <= j < k <= l < s.length and both the substrings s[i...j] and s[k...l] are palindromes and have odd lengths. s[i...j] denotes a substring from index i to index j inclusive. Return the maximum possible product of the lengths of the two non-intersecting palindromic substrings. A palindrome is a string that is the same forward and backward. A substring is a contiguous sequence of characters in a string. Example 1: Input: s = "ababbb" Output: 9 Explanation: Substrings "aba" and "bbb" are palindromes with odd length. product = 3 * 3 = 9.

Example 2: Input: s = "zaaaxbbby" Output: 9 Explanation: Substrings "aaa" and "bbb" are palindromes with odd length. product = 3 * 3 = 9.

Constraints:

2 <= s.length <= 105 s consists of lowercase English letters.
==
Problem Number: 2008 URL: https://leetcode.com/problems/find-the-longest-valid-obstacle-course-at-each-position Title: 1964. Find the Longest Valid Obstacle Course at Each Position Problem Description: You want to build some obstacle courses. You are given a 0-indexed integer array obstacles of length n, where obstacles[i] describes the height of the ith obstacle. For every index i between 0 and n - 1 (inclusive), find the length of the longest obstacle course in obstacles such that:

You choose any number of obstacles between 0 and i inclusive. You must include the ith obstacle in the course. You must put the chosen obstacles in the same order as they appear in obstacles. Every obstacle (except the first) is taller than or the same height as the obstacle immediately before it.

Return an array ans of length n, where ans[i] is the length of the longest obstacle course for index i as described above. Example 1: Input: obstacles = [1,2,3,2] Output: [1,2,3,3] Explanation: The longest valid obstacle course at each position is: - i = 0: [1], [1] has length 1. - i = 1: [1,2], [1,2] has length 2. - i = 2: [1,2,3], [1,2,3] has length 3. - i = 3: [1,2,3,2], [1,2,2] has length 3.

Example 2: Input: obstacles = [2,2,1] Output: [1,2,1] Explanation: The longest valid obstacle course at each position is: - i = 0: [2], [2] has length 1. - i = 1: [2,2], [2,2] has length 2. - i = 2: [2,2,1], [1] has length 1.

Example 3: Input: obstacles = [3,1,5,6,4,2] Output: [1,1,2,3,2,2] Explanation: The longest valid obstacle course at each position is: - i = 0: [3], [3] has length 1. - i = 1: [3,1], [1] has length 1. - i = 2: [3,1,5], [3,5] has length 2. [1,5] is also valid. - i = 3: [3,1,5,6], [3,5,6] has length 3. [1,5,6] is also valid. - i = 4: [3,1,5,6,4], [3,4] has length 2. [1,4] is also valid. - i = 5: [3,1,5,6,4,2], [1,2] has length 2.

Constraints:

n == obstacles.length 1 <= n <= 105 1 <= obstacles[i] <= 107

==

Problem Number: 2009 URL: https://leetcode.com/problems/last-day-where-you-can-still-cross Title: 1970. Last Day Where You Can Still Cross Problem Description: There is a 1-based binary matrix where 0 represents land and 1 represents water. You are given integers row and col representing the number of rows and columns in the matrix, respectively. Initially on day 0, the entire matrix is land. However, each day a new cell becomes flooded with water. You are given a 1-based 2D array cells, where cells[i] = [ri, ci] represents that on the ith day, the cell on the rith row and cith column (1-based coordinates) will be covered with water (i.e., changed to 1). You want to find the last day that it is possible to walk from the top to the bottom by only walking on land cells. You can start from any cell in the top row and end at any cell in the bottom row. You can only travel in the four cardinal directions (left, right, up, and down). Return the last day where it is possible to walk from the top to the bottom by only walking on land cells.   Example 1:

Input: row = 2, col = 2, cells = [[1,1],[2,1],[1,2],[2,2]] Output: 2 Explanation: The above image depicts how the matrix changes each day starting from day 0. The last day where it is possible to cross from top to bottom is on day 2.

Example 2:

Input: row = 2, col = 2, cells = [[1,1],[1,2],[2,1],[2,2]] Output: 1 Explanation: The above image depicts how the matrix changes each day starting from day 0. The last day where it is possible to cross from top to bottom is on day 1.

Example 3:

Input: row = 3, col = 3, cells = [[1,2],[2,1],[3,3],[2,2],[1,1],[1,3],[2,3],[3,2],[3,1]] Output: 3 Explanation: The above image depicts how the matrix changes each day starting from day 0. The last day where it is possible to cross from top to bottom is on day 3.

Constraints:

2 <= row, col <= 2 * 104 4 <= row * col <= 2 * 104 cells.length == row * col 1 <= ri <= row 1 <= ci <= col All the values of cells are unique.

==

Problem Number: 2010 URL: https://leetcode.com/problems/number-of-ways-to-separate-numbers Title: 1977. Number of Ways to Separate Numbers Problem Description: You wrote down many positive integers in a string called num. However, you realized that you forgot to add commas to seperate the different numbers. You remember that the list of integers was non-decreasing and that no integer had leading zeros. Return the number of possible lists of integers that you could have written down to get the string num. Since the answer may be large, return it modulo 109 + 7. Example 1: Input: num = "327" Output: 2 Explanation: You could have written down the numbers: 3, 27 327

Example 2: Input: num = "094" Output: 0 Explanation: No numbers can have leading zeros and all numbers must be positive.

Example 3: Input: num = "0" Output: 0 Explanation: No numbers can have leading zeros and all numbers must be positive.

Constraints:

1 <= num.length <= 3500 num consists of digits '0' through '9'.

==

Problem Number: 2011 URL: https://leetcode.com/problems/find-array-given-subset-sums Title: 1982. Find Array Given Subset Sums Problem Description: You are given an integer n representing the length of an unknown array that you are trying to recover. You are also given an array sums containing the values of all 2n subset sums of the unknown array (in no particular order). Return the array ans of length n representing the unknown array. If multiple answers exist, return any of them. An array sub is a subset of an array arr if sub can be obtained from arr by deleting some (possibly zero or all) elements of arr. The sum of the elements in sub is one possible subset sum of arr. The sum of an empty array is considered to be 0. Note: Test cases are generated such that there will always be at least one correct answer. Example 1: Input: n = 3, sums = [-3,-2,-1,0,0,1,2,3] Output: [1,2,-3] Explanation: [1,2,-3] is able to achieve the given subset sums: - []: sum is 0 - [1]: sum is 1 - [2]: sum is 2 - [1,2]: sum is 3 - [-3]: sum is -3 - [1,-3]: sum is -2 - [2,-3]: sum is -1 - [1,2,-3]: sum is 0 Note that any permutation of [1,2,-3] and also any permutation of [-1,-2,3] will also be accepted.

Example 2: Input: n = 2, sums = [0,0,0,0] Output: [0,0] Explanation: The only correct answer is [0,0].

Example 3: Input: n = 4, sums = [0,0,5,5,4,-1,4,9,9,-1,4,3,4,8,3,8] Output: [0,-1,4,5] Explanation: [0,-1,4,5] is able to achieve the given subset sums.

Constraints:

1 <= n <= 15 sums.length == 2n -104 <= sums[i] <= 104

==

Problem Number: 2012 URL: https://leetcode.com/problems/number-of-unique-good-subsequences Title: 1987. Number of Unique Good Subsequences Problem Description: You are given a binary string binary. A subsequence of binary is considered good if it is not empty and has no leading zeros (with the exception of "0"). Find the number of unique good subsequences of binary.

For example, if binary = "001", then all the good subsequences are ["0", "0", "1"], so the unique good subsequences are "0" and "1". Note that subsequences "00", "01", and "001" are not good because they have leading zeros.

Return the number of unique good subsequences of binary. Since the answer may be very large, return it modulo 109 + 7. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: binary = "001" Output: 2 Explanation: The good subsequences of binary are ["0", "0", "1"]. The unique good subsequences are "0" and "1".

Example 2: Input: binary = "11" Output: 2 Explanation: The good subsequences of binary are ["1", "1", "11"]. The unique good subsequences are "1" and "11". Example 3: Input: binary = "101" Output: 5 Explanation: The good subsequences of binary are ["1", "0", "1", "10", "11", "101"]. The unique good subsequences are "0", "1", "10", "11", and "101".

Constraints:

1 <= binary.length <= 105 binary consists of only '0's and '1's.

==

Problem Number: 2013 URL: https://leetcode.com/problems/the-number-of-good-subsets Title: 1994. The Number of Good Subsets Problem Description: You are given an integer array nums. We call a subset of nums good if its product can be represented as a product of one or more distinct prime numbers.

For example, if nums = [1, 2, 3, 4]:

[2, 3], [1, 2, 3], and [1, 3] are good subsets with products 6 = 2*3, 6 = 2*3, and 3 = 3 respectively. [1, 4] and [4] are not good subsets with products 4 = 2*2 and 4 = 2*2 respectively.

Return the number of different good subsets in nums modulo 109 + 7. A subset of nums is any array that can be obtained by deleting some (possibly none or all) elements from nums. Two subsets are different if and only if the chosen indices to delete are different. Example 1: Input: nums = [1,2,3,4] Output: 6 Explanation: The good subsets are: - [1,2]: product is 2, which is the product of distinct prime 2. - [1,2,3]: product is 6, which is the product of distinct primes 2 and 3. - [1,3]: product is 3, which is the product of distinct prime 3. - [2]: product is 2, which is the product of distinct prime 2. - [2,3]: product is 6, which is the product of distinct primes 2 and 3. - [3]: product is 3, which is the product of distinct prime 3.

Example 2: Input: nums = [4,2,3,15] Output: 5 Explanation: The good subsets are: - [2]: product is 2, which is the product of distinct prime 2. - [2,3]: product is 6, which is the product of distinct primes 2 and 3. - [2,15]: product is 30, which is the product of distinct primes 2, 3, and 5. - [3]: product is 3, which is the product of distinct prime 3. - [15]: product is 15, which is the product of distinct primes 3 and 5.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 30

==
Problem Number: 2014 URL: https://leetcode.com/problems/gcd-sort-of-an-array Title: 1998. GCD Sort of an Array Problem Description: You are given an integer array nums, and you can perform the following operation any number of times on nums:

Swap the positions of two elements nums[i] and nums[j] if gcd(nums[i], nums[j]) > 1 where gcd(nums[i], nums[j]) is the greatest common divisor of nums[i] and nums[j].

Return true if it is possible to sort nums in non-decreasing order using the above swap method, or false otherwise.   Example 1: Input: nums = [7,21,3] Output: true Explanation: We can sort [7,21,3] by performing the following operations: - Swap 7 and 21 because gcd(7,21) = 7. nums = [21,7,3] - Swap 21 and 3 because gcd(21,3) = 3. nums = [3,7,21]

Example 2: Input: nums = [5,2,6,2] Output: false Explanation: It is impossible to sort the array because 5 cannot be swapped with any other element.

Example 3: Input: nums = [10,5,9,3,15] Output: true We can sort [10,5,9,3,15] by performing the following operations: - Swap 10 and 15 because gcd(10,15) = 5. nums = [15,5,9,3,10] - Swap 15 and 3 because gcd(15,3) = 3. nums = [3,5,9,15,10] - Swap 10 and 15 because gcd(10,15) = 5. nums = [3,5,9,10,15]

Constraints:

1 <= nums.length <= 3 * 104 2 <= nums[i] <= 105

==
Problem Number: 2015 URL: https://leetcode.com/problems/smallest-missing-genetic-value-in-each-subtree Title: 2003. Smallest Missing Genetic Value in Each Subtree Problem Description: There is a family tree rooted at 0 consisting of n nodes numbered 0 to n - 1. You are given a 0-indexed integer array parents, where parents[i] is the parent for node i. Since node 0 is the root, parents[0] == -1. There are 105 genetic values, each represented by an integer in the inclusive range [1, 105]. You are given a 0-indexed integer array nums, where nums[i] is a distinct genetic value for node i. Return an array ans of length n where ans[i] is the smallest genetic value that is missing from the subtree rooted at node i. The subtree rooted at a node x contains node x and all of its descendant nodes.   Example 1:

Input: parents = [-1,0,0,2], nums = [1,2,3,4] Output: [5,1,1,1] Explanation: The answer for each subtree is calculated as follows: - 0: The subtree contains nodes [0,1,2,3] with values [1,2,3,4]. 5 is the smallest missing value. - 1: The subtree contains only node 1 with value 2. 1 is the smallest missing value. - 2: The subtree contains nodes [2,3] with values [3,4]. 1 is the smallest missing value.

- 3: The subtree contains only node 3 with value 4. 1 is the smallest missing value.

Example 2:

Input: parents = [-1,0,1,0,3,3], nums = [5,4,6,2,1,3] Output: [7,1,1,4,2,1] Explanation: The answer for each subtree is calculated as follows: - 0: The subtree contains nodes [0,1,2,3,4,5] with values [5,4,6,2,1,3]. 7 is the smallest missing value. - 1: The subtree contains nodes [1,2] with values [4,6]. 1 is the smallest missing value. - 2: The subtree contains only node 2 with value 6. 1 is the smallest missing value. - 3: The subtree contains nodes [3,4,5] with values [2,1,3]. 4 is the smallest missing value. - 4: The subtree contains only node 4 with value 1. 2 is the smallest missing value. - 5: The subtree contains only node 5 with value 3. 1 is the smallest missing value.

Example 3: Input: parents = [-1,2,3,0,2,4,1], nums = [2,3,4,5,6,7,8] Output: [1,1,1,1,1,1,1] Explanation: The value 1 is missing from all the subtrees.

Constraints:

n == parents.length == nums.length 2 <= n <= 105 0 <= parents[i] <= n - 1 for i != 0 parents[0] == -1 parents represents a valid tree. 1 <= nums[i] <= 105 Each nums[i] is distinct.

==
Problem Number: 2016 URL: https://leetcode.com/problems/minimum-number-of-operations-to-make-array-continuous Title: 2009. Minimum Number of Operations to Make Array Continuous Problem Description: You are given an integer array nums. In one operation, you can replace any element in nums with any integer. nums is considered continuous if both of the following conditions are fulfilled:

All elements in nums are unique. The difference between the maximum element and the minimum element in nums equals nums.length - 1.

For example, nums = [4, 2, 5, 3] is continuous, but nums = [1, 2, 3, 5, 6] is not continuous. Return the minimum number of operations to make nums continuous. Example 1: Input: nums = [4,2,5,3] Output: 0 Explanation: nums is already continuous.

Example 2: Input: nums = [1,2,3,5,6] Output: 1 Explanation: One possible solution is to change the last element to 4. The resulting array is [1,2,3,5,4], which is continuous.

Example 3: Input: nums = [1,10,100,1000] Output: 3 Explanation: One possible solution is to: - Change the second element to 2. - Change the third element to 3. - Change the fourth element to 4. The resulting array is [1,2,3,4], which is continuous.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 2017 URL: https://leetcode.com/problems/longest-subsequence-repeated-k-times Title: 2014. Longest Subsequence Repeated k Times Problem Description: You are given a string s of length n, and an integer k. You are tasked to find the longest subsequence repeated k times in string s. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. A subsequence seq is repeated k times in the string s if seq * k is a subsequence of s, where seq * k represents a string constructed by concatenating seq k times.

For example, "bba" is repeated 2 times in the string "bababcba", because the string "bbabba", constructed by concatenating "bba" 2 times, is a subsequence of the string "bababcba".

Return the longest subsequence repeated k times in string s. If multiple such subsequences are found, return the lexicographically largest one. If there is no such subsequence, return an empty string. Example 1:

Input: s = "letsleetcode", k = 2 Output: "let" Explanation: There are two longest subsequences repeated 2 times: "let" and "ete". "let" is the lexicographically largest one.

Example 2: Input: s = "bb", k = 2 Output: "b" Explanation: The longest subsequence repeated 2 times is "b".

Example 3: Input: s = "ab", k = 2 Output: "" Explanation: There is no subsequence repeated 2 times. Empty string is returned.

Constraints:

n == s.length 2 <= n, k <= 2000 2 <= n < k * 8 s consists of lowercase English letters.

==
Problem Number: 2018 URL: https://leetcode.com/problems/the-score-of-students-solving-math-expression Title: 2019. The Score of Students Solving Math Expression Problem Description: You are given a string s that contains digits 0-9, addition symbols '+', and multiplication symbols '*' only, representing a valid math expression of single digit numbers (e.g., 3+5*2). This expression was given to n elementary school students. The students were instructed to get the answer of the expression by following this order of operations:

Compute multiplication, reading from left to right; Then, Compute addition, reading from left to right.

You are given an integer array answers of length n, which are the submitted answers of the students in no particular order. You are asked to grade the answers, by following these rules:

If an answer equals the correct answer of the expression, this student will be rewarded 5 points; Otherwise, if the answer could be interpreted as if the student applied the operators in the wrong order but had correct arithmetic, this student will be rewarded 2 points; Otherwise, this student will be rewarded 0 points.

Return the sum of the points of the students. Example 1:

Input: s = "7+3*1*2", answers = [20,13,42] Output: 7 Explanation: As illustrated above, the correct answer of the expression is 13, therefore one student is rewarded 5 points: [20,13,42] A student might have applied the operators in this wrong order: ((7+3)*1)*2 = 20. Therefore one student is rewarded 2 points: [20,13,42] The points for the students are: [2,5,0]. The sum of the points is 2+5+0=7.

Example 2: Input: s = "3+5*2", answers = [13,0,10,13,13,16,16] Output: 19 Explanation: The correct answer of the expression is 13, therefore three students are rewarded 5 points each: [13,0,10,13,13,16,16] A student might have applied the operators in this wrong order: ((3+5)*2 = 16. Therefore two students are rewarded 2 points: [13,0,10,13,13,16,16] The points for the students are: [5,0,0,5,5,2,2]. The sum of the points is 5+0+0+5+5+2+2=19.

Example 3: Input: s = "6+0*1", answers = [12,9,6,4,8,6] Output: 10 Explanation: The correct answer of the expression is 6. If a student had incorrectly done (6+0)*1, the answer would also be 6. By the rules of grading, the students will still be rewarded 5 points (as they got the correct answer), not 2 points. The points for the students are: [0,0,5,0,0,5]. The sum of the points is 10.

Constraints:

3 <= s.length <= 31 s represents a valid expression that contains only digits 0-9, '+', and '*' only. All the integer operands in the expression are in the inclusive range [0, 9]. 1 <= The count of all operators ('+' and '*') in the math expression <= 15 Test data are generated such that the correct answer of the expression is in the range of [0, 1000]. n == answers.length 1 <= n <= 104 0 <= answers[i] <= 1000

==
Problem Number: 2019 URL: https://leetcode.com/problems/maximum-number-of-ways-to-partition-an-array Title: 2025. Maximum Number of Ways to Partition an Array Problem Description: You are given a 0-indexed integer array nums of length n. The number of ways to partition nums is the number of pivot indices that satisfy both conditions:

1 <= pivot < n nums[0] + nums[1] + ... + nums[pivot - 1] == nums[pivot] + nums[pivot + 1] + ... + nums[n - 1]

You are also given an integer k. You can choose to change the value of one element of nums to k, or to leave the array unchanged. Return the maximum possible number of ways to partition nums to satisfy both conditions after changing at most one element. Example 1: Input: nums = [2,-1,2], k = 3 Output: 1 Explanation: One optimal approach is to change nums[0] to k. The array becomes [3,-1,2]. There is one way to partition the array: - For pivot = 2, we have the partition [3,-1 | 2]: 3 + -1 == 2.

Example 2: Input: nums = [0,0,0], k = 1 Output: 2 Explanation: The optimal approach is to leave the array unchanged. There are two ways to partition the array: - For pivot = 1, we have the partition [0 | 0,0]: 0 == 0 + 0. - For pivot = 2, we have the partition [0,0 | 0]: 0 + 0 == 0.

Example 3: Input: nums = [22,4,-25,-20,-15,15,-16,7,19,-10,0,-13,-14], k = -33 Output: 4 Explanation: One optimal approach is to change nums[2] to k. The array becomes [22,4,-33,-20,-15,15,-16,7,19,-10,0,-13,-14]. There are four ways to partition the array.

Constraints:

n == nums.length 2 <= n <= 105 -105 <= k, nums[i] <= 105

==
Problem Number: 2020 URL: https://leetcode.com/problems/smallest-k-length-subsequence-with-occurrences-of-a-letter Title: 2030. Smallest K-Length Subsequence With Occurrences of a Letter Problem Description: You are given a string s, an integer k, a letter letter, and an integer repetition. Return the lexicographically smallest subsequence of s of length k that has the letter letter appear at least repetition times. The test cases are generated so that the letter appears in s at least repetition times. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters. A string a is lexicographically smaller than a string b if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b. Example 1: Input: s = "leet", k = 3, letter = "e", repetition = 1 Output: "eet" Explanation: There are four subsequences of length 3 that have the letter 'e' appear at least 1 time: - "lee" (from "leet") - "let" (from "leet") - "let" (from "leet") - "eet" (from "leet") The lexicographically smallest subsequence among them is "eet".

Example 2:

Input: s = "leetcode", k = 4, letter = "e", repetition = 2 Output: "ecde" Explanation: "ecde" is the lexicographically smallest subsequence of length 4 that has the letter "e" appear at least 2 times.

Example 3: Input: s = "bb", k = 2, letter = "b", repetition = 2 Output: "bb" Explanation: "bb" is the only subsequence of length 2 that has the letter "b" appear at least 2 times.

Constraints:

1 <= repetition <= k <= s.length <= 5 * 104 s consists of lowercase English letters. letter is a lowercase English letter, and appears in s at least repetition times.

==

Problem Number: 2021 URL: https://leetcode.com/problems/partition-array-into-two-arrays-to-minimize-sum-difference Title: 2035. Partition Array Into Two Arrays to Minimize Sum Difference Problem Description: You are given an integer array nums of 2 * n integers. You need to partition nums into two arrays of length n to minimize the absolute difference of the sums of the arrays. To partition nums, put each element of nums into one of the two arrays. Return the minimum possible absolute difference.   Example 1:

Input: nums = [3,9,7,3] Output: 2 Explanation: One optimal partition is: [3,9] and [7,3]. The absolute difference between the sums of the arrays is abs((3 + 9) - (7 + 3)) = 2.

Example 2: Input: nums = [-36,36] Output: 72 Explanation: One optimal partition is: [-36] and [36]. The absolute difference between the sums of the arrays is abs((-36) - (36)) = 72.

Example 3:

Input: nums = [2,-1,0,4,-2,-9] Output: 0 Explanation: One optimal partition is: [2,4,-9] and [-1,0,-2]. The absolute difference between the sums of the arrays is abs((2 + 4 + -9) - (-1 + 0 + -2)) = 0.

Constraints:

1 <= n <= 15 nums.length == 2 * n -107 <= nums[i] <= 107

==

Problem Number: 2022 URL: https://leetcode.com/problems/kth-smallest-product-of-two-sorted-arrays Title: 2040. Kth Smallest Product of Two Sorted Arrays Problem Description: Given two sorted 0-indexed integer arrays nums1 and nums2 as well as an integer k, return the kth (1-based) smallest product of nums1[i] * nums2[j] where 0 <= i < nums1.length and 0 <= j < nums2.length. Example 1: Input: nums1 = [2,5], nums2 = [3,4], k = 2 Output: 8 Explanation: The 2 smallest products are: - nums1[0] * nums2[0] = 2 * 3 = 6 - nums1[0] * nums2[1] = 2 * 4 = 8 The 2nd smallest product is 8.

Example 2: Input: nums1 = [-4,-2,0,3], nums2 = [2,4], k = 6 Output: 0 Explanation: The 6 smallest products are: - nums1[0] * nums2[1] = (-4) * 4 = -16 - nums1[0] * nums2[0] = (-4) * 2 = -8 - nums1[1] * nums2[1] = (-2) * 4 = -8 - nums1[1] * nums2[0] = (-2) * 2 = -4 - nums1[2] * nums2[0] = 0 * 2 = 0 - nums1[2] * nums2[1] = 0 * 4 = 0 The 6th smallest product is 0.

Example 3: Input: nums1 = [-2,-1,0,1,2], nums2 = [-3,-1,2,4,5], k = 3 Output: -6 Explanation: The 3 smallest products are: - nums1[0] * nums2[4] = (-2) * 5 = -10 - nums1[0] * nums2[3] = (-2) * 4 = -8 - nums1[4] * nums2[0] = 2 * (-3) = -6 The 3rd smallest product is -6.

Constraints:

1 <= nums1.length, nums2.length <= 5 * 104 -105 <= nums1[i], nums2[j] <= 105 1 <= k <= nums1.length * nums2.length nums1 and nums2 are sorted.

==

Problem Number: 2023 URL: https://leetcode.com/problems/second-minimum-time-to-reach-destination Title: 2045. Second Minimum Time to Reach Destination Problem Description: A city is represented as a bi-directional connected graph with n vertices where each vertex is labeled from 1 to n (inclusive). The edges in the graph are represented as a 2D integer array edges, where each edges[i] = [ui, vi] denotes a bi-directional edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself. The time taken to traverse any edge is time minutes. Each vertex has a traffic signal which changes its color from green to red and vice versa every change minutes. All signals change at the same time. You can enter a vertex at any time, but can leave a vertex only when the signal is green. You cannot wait at a vertex if the signal is green. The second minimum value is defined as the smallest value strictly larger than the minimum value.

For example the second minimum value of [2, 3, 4] is 3, and the second minimum value of [2, 2, 4] is 4.

Given n, edges, time, and change, return the second minimum time it will take to go from vertex 1 to vertex n. Notes:

You can go through any vertex any number of times, including 1 and n. You can assume that when the journey starts, all signals have just turned green.

Example 1: Input: n = 5, edges = [[1,2],[1,3],[1,4],[3,4],[4,5]], time = 3, change = 5 Output: 13 Explanation: The figure on the left shows the given graph. The blue path in the figure on the right is the minimum time path. The time taken is: - Start at 1, time elapsed=0 - 1 -> 4: 3 minutes, time elapsed=3 - 4 -> 5: 3 minutes, time elapsed=6 Hence the minimum time needed is 6 minutes.

The red path shows the path to get the second minimum time. - Start at 1, time elapsed=0 - 1 -> 3: 3 minutes, time elapsed=3 - 3 -> 4: 3 minutes, time elapsed=6 - Wait at 4 for 4 minutes, time elapsed=10 - 4 -> 5: 3 minutes, time elapsed=13 Hence the second minimum time is 13 minutes.

Example 2:

Input: n = 2, edges = [[1,2]], time = 3, change = 2 Output: 11 Explanation: The minimum time path is 1 -> 2 with time = 3 minutes. The second minimum time path is 1 -> 2 -> 1 -> 2 with time = 11 minutes. Constraints:

2 <= n <= 104 n - 1 <= edges.length <= min(2 * 104, n * (n - 1) / 2) edges[i].length == 2 1 <= ui, vi <= n ui != vi There are no duplicate edges. Each vertex can be reached directly or indirectly from every other vertex. 1 <= time, change <= 103

==
Problem Number: 2024 URL: https://leetcode.com/problems/parallel-courses-iii Title: 2050. Parallel Courses III Problem Description: You are given an integer n, which indicates that there are n courses labeled from 1 to n. You are also given a 2D integer array relations where relations[j] = [prevCoursej, nextCoursej] denotes that course prevCoursej has to be completed before course nextCoursej (prerequisite relationship). Furthermore, you are given a 0-indexed integer array time where time[i] denotes how many months it takes to complete the (i+1)th course. You must find the minimum number of months needed to complete all the courses following these rules:

You may start taking a course at any time if the prerequisites are met. Any number of courses can be taken at the same time.

Return the minimum number of months needed to complete all the courses. Note: The test cases are generated such that it is possible to complete every course (i.e., the graph is a directed acyclic graph).   Example 1:

Input: n = 3, relations = [[1,3],[2,3]], time = [3,2,5] Output: 8 Explanation: The figure above represents the given graph and the time required to complete each course. We start course 1 and course 2 simultaneously at month 0. Course 1 takes 3 months and course 2 takes 2 months to complete respectively. Thus, the earliest time we can start course 3 is at month 3, and the total time required is 3 + 5 = 8 months.

Example 2:

Input: n = 5, relations = [[1,5],[2,5],[3,5],[3,4],[4,5]], time = [1,2,3,4,5] Output: 12 Explanation: The figure above represents the given graph and the time required to complete each course. You can start courses 1, 2, and 3 at month 0. You can complete them after 1, 2, and 3 months respectively. Course 4 can be taken only after course 3 is completed, i.e., after 3 months. It is completed after 3 + 4 = 7 months. Course 5 can be taken only after courses 1, 2, 3, and 4 have been completed, i.e., after max(1,2,3,7) = 7 months. Thus, the minimum time needed to complete all the courses is 7 + 5 = 12 months.

Constraints:

1 <= n <= 5 * 104 0 <= relations.length <= min(n * (n - 1) / 2, 5 * 104) relations[j].length == 2 1 <= prevCoursej, nextCoursej <= n prevCoursej != nextCoursej All the pairs [prevCoursej, nextCoursej] are unique. time.length == n 1 <= time[i] <= 104 The given graph is a directed acyclic graph.

==
Problem Number: 2025 URL: https://leetcode.com/problems/number-of-valid-move-combinations-on-chessboard Title: 2056. Number of Valid Move Combinations On Chessboard Problem Description: There is an 8 x 8 chessboard containing n pieces (rooks, queens, or bishops). You are given a string array pieces of length n, where pieces[i] describes the type (rook, queen, or bishop) of the ith piece. In addition, you are given a 2D integer array positions

also of length n, where positions[i] = [ri, ci] indicates that the ith piece is currently at the 1-based coordinate (ri, ci) on the chessboard. When making a move for a piece, you choose a destination square that the piece will travel toward and stop on.

A rook can only travel horizontally or vertically from (r, c) to the direction of (r+1, c), (r-1, c), (r, c+1), or (r, c-1). A queen can only travel horizontally, vertically, or diagonally from (r, c) to the direction of (r+1, c), (r-1, c), (r, c+1), (r, c-1), (r+1, c+1), (r+1, c-1), (r-1, c+1), (r-1, c-1). A bishop can only travel diagonally from (r, c) to the direction of (r+1, c+1), (r+1, c-1), (r-1, c+1), (r-1, c-1).

You must make a move for every piece on the board simultaneously. A move combination consists of all the moves performed on all the given pieces. Every second, each piece will instantaneously travel one square towards their destination if they are not already at it. All pieces start traveling at the 0th second. A move combination is invalid if, at a given time, two or more pieces occupy the same square. Return the number of valid move combinations. Notes:

No two pieces will start in the same square. You may choose the square a piece is already on as its destination. If two pieces are directly adjacent to each other, it is valid for them to move past each other and swap positions in one second.

Example 1:

Input: pieces = [”rook”], positions = [[1,1]] Output: 15 Explanation: The image above shows the possible squares the piece can move to.

Example 2:

Input: pieces = [”queen”], positions = [[1,1]] Output: 22 Explanation: The image above shows the possible squares the piece can move to.

Example 3:

Input: pieces = [”bishop”], positions = [[4,3]] Output: 12 Explanation: The image above shows the possible squares the piece can move to.

Constraints:

n == pieces.length n == positions.length 1 <= n <= 4 pieces only contains the strings ”rook”, ”queen”, and ”bishop”. There will be at most one queen on the chessboard. 1 <= xi, yi <= 8 Each positions[i] is distinct.

==

Problem Number: 2026 URL: https://leetcode.com/problems/check-if-an-original-string-exists-given-two-encoded-strings Title: 2060. Check if an Original String Exists Given Two Encoded Strings Problem Description: An original string, consisting of lowercase English letters, can be encoded by the following steps:

Arbitrarily split it into a sequence of some number of non-empty substrings. Arbitrarily choose some elements (possibly none) of the sequence, and replace each with its length (as a numeric string). Concatenate the sequence as the encoded string.

For example, one way to encode an original string "abcdefghijklmnop" might be:

Split it as a sequence: ["ab", "cdefghijklmn", "o", "p"]. Choose the second and third elements to be replaced by their lengths, respectively. The sequence becomes ["ab", "12", "1", "p"]. Concatenate the elements of the sequence to get the encoded string: "ab121p".

Given two encoded strings s1 and s2, consisting of lowercase English letters and digits 1-9 (inclusive), return true if there exists an original string that could be encoded as both s1 and s2. Otherwise, return false. Note: The test cases are generated such that the number of consecutive digits in s1 and s2 does not exceed 3. Example 1: Input: s1 = "internationalization", s2 = "i18n" Output: true Explanation: It is possible that "internationalization" was the original string. - "internationalization" -> Split: ["internationalization"] -> Do not replace any element -> Concatenate: "internationalization", which is s1. - "internationalization" -> Split: ["i", "nternationalizatio", "n"] -> Replace: ["i", "18", "n"] -> Concatenate: "i18n", which is s2

Example 2: Input: s1 = "l123e", s2 = "44" Output: true Explanation: It is possible that "leetcode" was the original string. - "leetcode" -> Split: ["l", "e", "et", "cod", "e"] -> Replace: ["l", "1", "2", "3", "e"] -> Concatenate: "l123e", which is s1. - "leetcode" -> Split: ["leet", "code"] -> Replace: ["4", "4"] -> Concatenate: "44", which is s2.

Example 3: Input: s1 = "a5b", s2 = "c5b" Output: false Explanation: It is impossible. - The original string encoded as s1 must start with the letter 'a'. - The original string encoded as s2 must start with the letter 'c'.

Constraints:

1 <= s1.length, s2.length <= 40 s1 and s2 consist of digits 1-9 (inclusive), and lowercase English letters only. The number of consecutive digits in s1 and s2 does not exceed 3.

==

Problem Number: 2027 URL: https://leetcode.com/problems/maximum-path-quality-of-a-graph Title: 2065. Maximum Path Quality of a Graph Problem Description: There is an undirected graph with n nodes numbered from 0 to n - 1 (inclusive). You are given a 0-indexed integer array values where values[i] is the value of the ith node. You are also given a 0-indexed 2D integer array edges, where each edges[j] = [uj, vj, timej] indicates that there is an undirected edge between the nodes uj and vj, and it takes timej seconds to travel between the two nodes. Finally, you are given an integer maxTime. A valid path in the graph is any path that starts at node 0, ends at node 0, and takes at most maxTime seconds to complete. You may visit the same node multiple times. The quality of a valid path is the sum of the values of the unique nodes visited in the path (each node's value is added at most once to the sum). Return the maximum quality of a valid path. Note: There are at most four edges connected to each node. Example 1:

Input: values = [0,32,10,43], edges = [[0,1,10],[1,2,15],[0,3,10]], maxTime = 49 Output: 75 Explanation: One possible path is 0 -> 1 -> 0 -> 3 -> 0. The total time taken is 10 + 10 + 10 + 10 = 40 <= 49. The nodes visited are 0, 1, and 3, giving a maximal path quality of 0 + 32 + 43 = 75.

Example 2:

Input: values = [5,10,15,20], edges = [[0,1,10],[1,2,10],[0,3,10]], maxTime = 30 Output: 25 Explanation: One possible path is 0 -> 3 -> 0. The total time taken is 10 + 10 = 20 <= 30. The nodes visited are 0 and 3, giving a maximal path quality of 5 + 20 = 25.

Example 3:

Input: values = [1,2,3,4], edges = [[0,1,10],[1,2,11],[2,3,12],[1,3,13]], maxTime = 50 Output: 7 Explanation: One possible path is 0 -> 1 -> 3 -> 1 -> 0. The total time taken is 10 + 13 + 13 + 10 = 46 <= 50. The nodes visited are 0, 1, and 3, giving a maximal path quality of 1 + 2 + 4 = 7.

Constraints:

n == values.length 1 <= n <= 1000 0 <= values[i] <= 108 0 <= edges.length <= 2000 edges[j].length == 3 0 <= uj < vj <= n - 1 10 <= timej, maxTime <= 100 All the pairs [uj, vj] are unique. There are at most four edges connected to each node. The graph may not be connected.

==

Problem Number: 2028 URL: https://leetcode.com/problems/maximum-number-of-tasks-you-can-assign Title: 2071. Maximum Number of Tasks You Can Assign Problem Description: You have n tasks and m workers. Each task has a strength requirement stored in a 0-indexed integer array tasks, with the ith task requiring tasks[i] strength to complete. The strength of each worker is stored in a 0-indexed integer array workers, with the jth worker having workers[j] strength. Each worker can only be assigned to a single task and must have a strength greater than or equal to the task's strength requirement (i.e., workers[j] >= tasks[i]). Additionally, you have pills magical pills that will increase a worker's strength by strength. You can decide which workers receive the magical pills, however, you may only give each worker at most one magical pill. Given the 0-indexed integer arrays tasks and workers and the integers pills and strength, return the maximum number of tasks that can be completed. Example 1: Input: tasks = [3,2,1], workers = [0,3,3], pills = 1, strength = 1 Output: 3 Explanation: We can assign the magical pill and tasks as follows: - Give the magical pill to worker 0. - Assign worker 0 to task 2 (0 + 1 >= 1) - Assign worker 1 to task 1 (3 >= 2) - Assign worker 2 to task 0 (3 >= 3)

Example 2: Input: tasks = [5,4], workers = [0,0,0], pills = 1, strength = 5 Output: 1 Explanation: We can assign the magical pill and tasks as follows: - Give the magical pill to worker 0. - Assign worker 0 to task 0 (0 + 5 >= 5)

Example 3: Input: tasks = [10,15,30], workers = [0,10,10,10,10], pills = 3, strength = 10 Output: 2 Explanation: We can assign the magical pills and tasks as follows: - Give the magical pill to worker 0 and worker 1. - Assign worker 0 to task 0 (0 + 10 >= 10) - Assign worker 1 to task 1 (10 + 10 >= 15) The last pill is not given because it will not make any worker strong enough for the last task.

Constraints:

n == tasks.length m == workers.length 1 <= n, m <= 5 * 104 0 <= pills <= m 0 <= tasks[i], workers[j], strength <= 109

==
Problem Number: 2029 URL: https://leetcode.com/problems/process-restricted-friend-requests Title: 2076. Process Restricted Friend Requests Problem Description: You are given an integer n indicating the number of people in a network. Each person is labeled from 0 to n - 1. You are also given a 0-indexed 2D integer array restrictions, where restrictions[i] = [xi, yi] means that person xi and person yi cannot become friends, either directly or indirectly through other people. Initially, no one is friends with each other. You are given a list of friend requests as a 0-indexed 2D integer array requests, where requests[j] = [uj, vj] is a friend request between person uj and person vj. A friend request is successful if uj and vj can be friends. Each friend request is processed in the given order (i.e., requests[j] occurs before requests[j + 1]), and upon a successful request, uj and vj become direct friends for all future friend requests. Return a boolean array result, where each result[j] is true if the jth friend request is successful or false if it is not. Note: If uj and vj are already direct friends, the request is still successful. Example 1: Input: n = 3, restrictions = [[0,1]], requests = [[0,2],[2,1]] Output: [true,false] Explanation: Request 0: Person 0 and person 2 can be friends, so they become direct friends. Request 1: Person 2 and person 1 cannot be friends since person 0 and person 1 would be indirect friends (1--2--0).

Example 2: Input: n = 3, restrictions = [[0,1]], requests = [[1,2],[0,2]] Output: [true,false] Explanation: Request 0: Person 1 and person 2 can be friends, so they become direct friends. Request 1: Person 0 and person 2 cannot be friends since person 0 and person 1 would be indirect friends (0--2--1).

Example 3: Input: n = 5, restrictions = [[0,1],[1,2],[2,3]], requests = [[0,4],[1,2],[3,1],[3,4]] Output: [true,false,true,false] Explanation: Request 0: Person 0 and person 4 can be friends, so they become direct friends. Request 1: Person 1 and person 2 cannot be friends since they are directly restricted. Request 2: Person 3 and person 1 can be friends, so they become direct friends. Request 3: Person 3 and person 4 cannot be friends since person 0 and person 1 would be indirect friends (0--4--3--1).

Constraints:

2 <= n <= 1000 0 <= restrictions.length <= 1000 restrictions[i].length == 2 0 <= xi, yi <= n - 1 xi != yi 1 <= requests.length <= 1000 requests[j].length == 2 0 <= uj, vj <= n - 1 uj != vj

==
Problem Number: 2030 URL: https://leetcode.com/problems/sum-of-k-mirror-numbers Title: 2081. Sum of k-Mirror Numbers Problem Description: A k-mirror number is a positive integer without leading zeros that reads the same both forward and backward in base-10 as well as in base-k.

For example, 9 is a 2-mirror number. The representation of 9 in base-10 and base-2 are 9 and 1001 respectively, which read the same both forward and backward. On the contrary, 4 is not a 2-mirror number. The representation of 4 in base-2 is 100, which does not read the same both forward and backward.

Given the base k and the number n, return the sum of the n smallest k-mirror numbers. Example 1: Input: k = 2, n = 5 Output: 25 Explanation: The 5 smallest 2-mirror numbers and their representations in base-2 are listed as follows: base-10 base-2 1 1 3 11 5 101 7 111 9 1001 Their sum = 1 + 3 + 5 + 7 + 9 = 25.

Example 2: Input: k = 3, n = 7 Output: 499 Explanation: The 7 smallest 3-mirror numbers are and their representations in base-3 are listed as follows: base-10 base-3 1 1 2 2 4 11 8 22 121 11111 151 12121 212 21212 Their sum = 1 + 2 + 4 + 8 + 121 + 151 + 212 = 499.

Example 3: Input: k = 7, n = 17 Output: 20379000 Explanation: The 17 smallest 7-mirror numbers are: 1, 2, 3, 4, 5, 6, 8, 121, 171, 242, 292, 16561, 65656, 2137312, 4602064, 6597956, 6958596

Constraints:

2 <= k <= 9 1 <= n <= 30

==
Problem Number: 2031 URL: https://leetcode.com/problems/count-fertile-pyramids-in-a-land Title: 2088. Count Fertile Pyramids in a Land Problem Description: A farmer has a rectangular grid of land with m rows and n columns that can be divided into unit cells. Each cell is either fertile (represented by a 1) or barren (represented by a 0). All cells outside the grid are considered barren. A pyramidal plot of land can be defined as a set of cells with the following criteria:

The number of cells in the set has to be greater than 1 and all cells must be fertile. The apex of a pyramid is the topmost cell of the pyramid. The height of a pyramid is the number of rows it covers. Let (r, c) be the apex of the pyramid, and its height be h. Then, the plot comprises of cells (i, j) where r <= i <= r + h - 1 and c - (i - r) <= j <= c + (i - r).

An inverse pyramidal plot of land can be defined as a set of cells with similar criteria:

The number of cells in the set has to be greater than 1 and all cells must be fertile. The apex of an inverse pyramid is the bottommost cell of the inverse pyramid. The height of an inverse pyramid is the number of rows it covers. Let (r, c) be the apex of the pyramid, and its height be h. Then, the plot comprises of cells (i, j) where r - h + 1 <= i <= r and c - (r - i) <= j <= c + (r - i).

Some examples of valid and invalid pyramidal (and inverse pyramidal) plots are shown below. Black cells indicate fertile cells.

Given a 0-indexed m x n binary matrix grid representing the farmland, return the total number of pyramidal and inverse pyramidal plots that can be found in grid.   Example 1:

Input: grid = [[0,1,1,0],[1,1,1,1]] Output: 2 Explanation: The 2 possible pyramidal plots are shown in blue and red respectively. There are no inverse pyramidal plots in this grid. Hence total number of pyramidal and inverse pyramidal plots is 2 + 0 = 2.

Example 2:

Input: grid = [[1,1,1],[1,1,1]] Output: 2 Explanation: The pyramidal plot is shown in blue, and the inverse pyramidal plot is shown in red. Hence the total number of plots is 1 + 1 = 2.

Example 3:

Input: grid = [[1,1,1,1,0],[1,1,1,1,1],[1,1,1,1,1],[0,1,0,0,1]] Output: 13 Explanation: There are 7 pyramidal plots, 3 of which are shown in the 2nd and 3rd figures. There are 6 inverse pyramidal plots, 2 of which are shown in the last figure. The total number of plots is 7 + 6 = 13.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 1000 1 <= m * n <= 105 grid[i][j] is either 0 or 1.

==

Problem Number: 2032 URL: https://leetcode.com/problems/find-all-people-with-secret Title: 2092. Find All People With Secret Problem Description: You are given an integer n indicating there are n people numbered from 0 to n - 1. You are also given a 0-indexed 2D integer array meetings where meetings[i] = [xi, yi, timei] indicates that person xi and person yi have a meeting at timei. A person may attend multiple meetings at the same time. Finally, you are given an integer firstPerson. Person 0 has a secret and initially shares the secret with a person firstPerson at time 0. This secret is then shared every time a meeting takes place with a person that has the secret. More formally, for every meeting, if a person xi has the secret at timei, then they will share the secret with person yi, and vice versa. The secrets are shared instantaneously.

That is, a person may receive the secret and share it with people in other meetings within the same time frame. Return a list of all the people that have the secret after all the meetings have taken place. You may return the answer in any order. Example 1: Input: n = 6, meetings = [[1,2,5],[2,3,8],[1,5,10]], firstPerson = 1 Output: [0,1,2,3,5] Explanation: At time 0, person 0 shares the secret with person 1. At time 5, person 1 shares the secret with person 2. At time 8, person 2 shares the secret with person 3. At time 10, person 1 shares the secret with person 5. Thus, people 0, 1, 2, 3, and 5 know the secret after all the meetings.

Example 2: Input: n = 4, meetings = [[3,1,3],[1,2,2],[0,3,3]], firstPerson = 3 Output: [0,1,3] Explanation: At time 0, person 0 shares the secret with person 3. At time 2, neither person 1 nor person 2 know the secret. At time 3, person 3 shares the secret with person 0 and person 1. Thus, people 0, 1, and 3 know the secret after all the meetings.

Example 3: Input: n = 5, meetings = [[3,4,2],[1,2,1],[2,3,1]], firstPerson = 1 Output: [0,1,2,3,4] Explanation: At time 0, person 0 shares the secret with person 1. At time 1, person 1 shares the secret with person 2, and person 2 shares the secret with person 3. Note that person 2 can share the secret at the same time as receiving it. At time 2, person 3 shares the secret with person 4. Thus, people 0, 1, 2, 3, and 4 know the secret after all the meetings.

Constraints:

2 <= n <= 105 1 <= meetings.length <= 105 meetings[i].length == 3 0 <= xi, yi <= n - 1 xi != yi 1 <= timei <= 105 1 <= firstPerson <= n - 1

==
Problem Number: 2033 URL: https://leetcode.com/problems/valid-arrangement-of-pairs Title: 2097. Valid Arrangement of Pairs Problem Description: You are given a 0-indexed 2D integer array pairs where pairs[i] = [starti, endi]. An arrangement of pairs is valid if for every index i where 1 <= i < pairs.length, we have endi-1 == starti. Return any valid arrangement of pairs. Note: The inputs will be generated such that there exists a valid arrangement of pairs. Example 1: Input: pairs = [[5,1],[4,5],[11,9],[9,4]] Output: [[11,9],[9,4],[4,5],[5,1]] Explanation: This is a valid arrangement since endi-1 always equals starti. end0 = 9 == 9 = start1 end1 = 4 == 4 = start2 end2 = 5 == 5 = start3

Example 2: Input: pairs = [[1,3],[3,2],[2,1]] Output: [[1,3],[3,2],[2,1]] Explanation: This is a valid arrangement since endi-1 always equals starti. end0 = 3 == 3 = start1 end1 = 2 == 2 = start2 The arrangements [[2,1],[1,3],[3,2]] and [[3,2],[2,1],[1,3]] are also valid.

Example 3: Input: pairs = [[1,2],[1,3],[2,1]] Output: [[1,2],[2,1],[1,3]] Explanation: This is a valid arrangement since endi-1 always equals starti. end0 = 2 == 2 = start1 end1 = 1 == 1 = start2

Constraints:

1 <= pairs.length <= 105 pairs[i].length == 2 0 <= starti, endi <= 109 starti != endi No two pairs are exactly the same. There exists a valid arrangement of pairs.

== Problem Number: 2034 URL: https://leetcode.com/problems/sequentially-ordinal-rank-tracker Title: 2102. Sequentially Ordinal Rank Tracker Problem Description: A scenic location is represented by its name and attractiveness score, where name is a unique string among all locations and score is an integer. Locations can be ranked from the best to the worst. The higher the score, the better the location. If the scores of two locations are equal, then the location with the lexicographically smaller name is better. You are building a system that tracks the ranking of locations with the system initially starting with no locations. It supports:

Adding scenic locations, one at a time. Querying the ith best location of all locations already added, where i is the number of times the system has been queried (including the current query).

For example, when the system is queried for the 4th time, it returns the 4th best location of all locations already added.

Note that the test data are generated so that at any time, the number of queries does not exceed the number of locations added to the system. Implement the SORTracker class:

SORTracker() Initializes the tracker system. void add(string name, int score) Adds a scenic location with name and score to the system. string get() Queries and returns the ith best location, where i is the number of times this method has been invoked (including this invocation).

Example 1: Input ["SORTracker", "add", "add", "get", "add", "get", "add", "get", "add", "get", "add", "get", "get"] [[], ["bradford", 2], ["branford", 3], [], ["alps", 2], [], ["orland", 2], [], ["orlando", 3], [], ["alpine", 2], [], []] Output [null, null, null, "branford", null, "alps", null, "bradford", null, "bradford", null, "bradford", "orland"]

Explanation SORTracker tracker = new SORTracker(); // Initialize the tracker system. tracker.add("bradford", 2); // Add location with name="bradford" and score=2 to the system. tracker.add("branford", 3); // Add location with name="branford" and score=3 to the system. tracker.get(); // The sorted locations, from best to worst, are: branford, bradford. // Note that branford precedes bradford due to its higher score (3 > 2). // This is the 1st time get() is called, so return the best location: "branford". tracker.add("alps", 2); // Add location with name="alps" and score=2 to the system. tracker.get(); // Sorted locations: branford, alps, bradford. // Note that alps precedes bradford even though they have the same score (2). // This is because "alps" is lexicographically smaller than "bradford". // Return the 2nd best location "alps", as it is the 2nd time get() is called. tracker.add("orland", 2); // Add location with name="orland" and score=2 to the system. tracker.get(); // Sorted locations: branford, alps, bradford, orland. // Return "bradford", as it is the 3rd time get() is called. tracker.add("orlando", 3); // Add location with name="orlando" and score=3 to the system. tracker.get(); // Sorted locations: branford, orlando, alps, bradford, orland. // Return "bradford". tracker.add("alpine", 2); // Add location with name="alpine" and score=2 to the system. tracker.get(); // Sorted locations: branford, orlando, alpine, alps, bradford, orland. // Return "bradford". tracker.get(); // Sorted locations: branford, orlando, alpine, alps, bradford, orland. // Return "orland".

Constraints:

name consists of lowercase English letters, and is unique among all locations. 1 <= name.length <= 10 1 <= score <= 105 At any time, the number of calls to get does not exceed the number of calls to add. At most 4 * 104 calls in total will be made to add and get.

==
Problem Number: 2035 URL: https://leetcode.com/problems/maximum-fruits-harvested-after-at-most-k-steps Title: 2106. Maximum Fruits Harvested After at Most K Steps Problem Description: Fruits are available at some positions on an infinite x-axis. You are given a 2D integer array fruits where fruits[i] = [positioni, amounti] depicts amounti fruits at the position positioni. fruits is already sorted by positioni in ascending order, and each positioni is unique. You are also given an integer startPos and an integer k. Initially, you are at the position startPos. From any position, you can either walk to the left or right. It takes one step to move one unit on the x-axis, and you can walk at most k steps in total. For every position you reach, you harvest all the fruits at that position, and the fruits will disappear from that position. Return the maximum total number of fruits you can harvest. Example 1:

Input: fruits = [[2,8],[6,3],[8,6]], startPos = 5, k = 4 Output: 9 Explanation: The optimal way is to: - Move right to position 6 and harvest 3 fruits - Move right to position 8 and harvest 6 fruits You moved 3 steps and harvested 3 + 6 = 9 fruits in total.

Example 2:

Input: fruits = [[0,9],[4,1],[5,7],[6,2],[7,4],[10,9]], startPos = 5, k = 4 Output: 14 Explanation: You can move at most k = 4 steps, so you cannot reach position 0 nor 10. The optimal way is to: - Harvest the 7 fruits at the starting position 5 - Move left to position 4 and harvest 1 fruit - Move right to position 6 and harvest 2 fruits - Move right to position 7 and harvest 4 fruits You moved 1 + 3 = 4 steps and harvested 7 + 1 + 2 + 4 = 14 fruits in total.

Example 3:

Input: fruits = [[0,3],[6,4],[8,5]], startPos = 3, k = 2 Output: 0 Explanation: You can move at most k = 2 steps and cannot reach any position with fruits.

Constraints:

1 <= fruits.length <= 105 fruits[i].length == 2 0 <= startPos, positioni <= 2 * 105 positioni-1 < positioni for any i > 0 (0-indexed) 1 <= amounti <= 104 0 <= k <= 2 * 105

== Problem Number: 2036 URL: https://leetcode.com/problems/minimum-operations-to-make-the-array-k-increasing Title: 2111. Minimum Operations to Make the Array K-Increasing Problem Description: You are given a 0-indexed array arr consisting of n positive integers, and a positive integer k. The array arr is called K-increasing if arr[i-k] <= arr[i] holds for every index i, where k <= i <= n-1.

For example, arr = [4, 1, 5, 2, 6, 2] is K-increasing for k = 2 because:

arr[0] <= arr[2] (4 <= 5) arr[1] <= arr[3] (1 <= 2) arr[2] <= arr[4] (5 <= 6) arr[3] <= arr[5] (2 <= 2)

However, the same arr is not K-increasing for k = 1 (because arr[0] > arr[1]) or k = 3 (because arr[0] > arr[3]).

In one operation, you can choose an index i and change arr[i] into any positive integer. Return the minimum number of operations required to make the array K-increasing for the given k. Example 1: Input: arr = [5,4,3,2,1], k = 1 Output: 4 Explanation: For k = 1, the resultant array has to be non-decreasing. Some of the K-increasing arrays that can be formed are [5,6,7,8,9], [1,1,1,1,1], [2,2,3,4,4]. All of them require 4 operations. It is suboptimal to change the array to, for example, [6,7,8,9,10] because it would take 5 operations. It can be shown that we cannot make the array K-increasing in less than 4 operations.

Example 2: Input: arr = [4,1,5,2,6,2], k = 2 Output: 0 Explanation: This is the same example as the one in the problem description. Here, for every index i where 2 <= i <= 5, arr[i-2] <= arr[i]. Since the given array is already K-increasing, we do not need to perform any operations. Example 3: Input: arr = [4,1,5,2,6,2], k = 3 Output: 2 Explanation: Indices 3 and 5 are the only ones not satisfying arr[i-3] <= arr[i] for 3 <= i <= 5. One of the ways we can make the array K-increasing is by changing arr[3] to 4 and arr[5] to 5. The array will now be [4,1,5,4,6,5]. Note that there can be other ways to make the array K-increasing, but none of them require less than 2 operations.

Constraints:

1 <= arr.length <= 105 1 <= arr[i], k <= arr.length

== Problem Number: 2037 URL: https://leetcode.com/problems/abbreviating-the-product-of-a-range Title: 2117. Abbreviating the Product of a Range Problem Description: You are given two positive integers left and right with left <= right. Calculate the product of all integers in the inclusive range [left, right]. Since the product may be very large, you will abbreviate it following these steps:

Count all trailing zeros in the product and remove them. Let us denote this count as C.

For example, there are 3 trailing zeros in 1000, and there are 0 trailing zeros in 546.

Denote the remaining number of digits in the product as d. If d > 10, then express the product as <pre>...<suf> where <pre> denotes the first 5 digits of the product, and <suf> denotes the last 5 digits of the product after removing all trailing zeros. If d <= 10, we keep it unchanged.

For example, we express 1234567654321 as 12345...54321, but 1234567 is represented as 1234567.

Finally, represent the product as a string "<pre>...<suf>eC".

For example, 12345678987600000 will be represented as "12345...89876e5".

Return a string denoting the abbreviated product of all integers in the inclusive range [left, right]. Example 1: Input: left = 1, right = 4 Output: "24e0" Explanation: The product is 1 × 2 × 3 × 4 = 24. There are no trailing zeros, so 24 remains the same. The abbreviation will end with "e0". Since the number of digits is 2, which is less than 10, we do not have to abbreviate it further. Thus, the final representation is "24e0".

Example 2: Input: left = 2, right = 11 Output: "399168e2" Explanation: The product is 39916800. There are 2 trailing zeros, which we remove to get 399168. The abbreviation will end with "e2". The number of digits after removing the trailing zeros is 6, so we do not abbreviate it further. Hence, the abbreviated product is "399168e2".

Example 3: Input: left = 371, right = 375 Output: "7219856259e3" Explanation: The product is 7219856259000.

Constraints:

1 <= left <= right <= 104

==

Problem Number: 2038 URL: https://leetcode.com/problems/recover-the-original-array Title: 2122. Recover the Original Array Problem Description: Alice had a 0-indexed array arr consisting of n positive integers. She chose an arbitrary positive integer k and created two new 0-indexed integer arrays lower and higher in the following manner:

lower[i] = arr[i] - k, for every index i where 0 <= i < n higher[i] = arr[i] + k, for every index i where 0 <= i < n

Unfortunately, Alice lost all three arrays. However, she remembers the integers that were present in the arrays lower and higher, but not the array each integer belonged to. Help Alice and recover the original array. Given an array nums consisting of 2n integers, where exactly n of the integers were present in lower and the remaining in higher, return the original array arr. In case the answer is not unique, return any valid array. Note: The test cases are generated such that there exists at least one valid array arr.   Example 1: Input: nums = [2,10,6,4,8,12] Output: [3,7,11] Explanation: If arr = [3,7,11] and k = 1, we get lower = [2,6,10] and higher = [4,8,12]. Combining lower and higher gives us [2,6,10,4,8,12], which is a permutation of nums. Another valid possibility is that arr = [5,7,9] and k = 3. In that case, lower = [2,4,6] and higher = [8,10,12].

Example 2: Input: nums = [1,1,3,3] Output: [2,2] Explanation: If arr = [2,2] and k = 1, we get lower = [1,1] and higher = [3,3]. Combining lower and higher gives us [1,1,3,3], which is equal to nums. Note that arr cannot be [1,3] because in that case, the only possible way to obtain [1,1,3,3] is with k = 0. This is invalid since k must be positive.

Example 3: Input: nums = [5,435] Output: [220] Explanation: The only possible combination is arr = [220] and k = 215. Using them, we get lower = [5] and higher = [435].

Constraints:

2 * n == nums.length 1 <= n <= 1000 1 <= nums[i] <= 109 The test cases are generated such that there exists at least one valid array arr.

==
Problem Number: 2039 URL: https://leetcode.com/problems/maximum-employees-to-be-invited-to-a-meeting Title: 2127. Maximum Employees to Be Invited to a Meeting Problem Description: A company is organizing a meeting and has a list of n employees, waiting to be invited. They have arranged for a large circular table, capable of seating any number of employees. The employees are numbered from 0 to n - 1. Each employee has a favorite person and they will attend the meeting only if they can sit next to their favorite person at the table. The favorite person of an employee is not themself. Given a 0-indexed integer array favorite, where favorite[i] denotes the favorite person of the ith employee, return the maximum number of employees that can be invited to the meeting.   Example 1:

Input: favorite = [2,2,1,2] Output: 3 Explanation: The above figure shows how the company can invite employees 0, 1, and 2, and seat them at the round table. All employees cannot be invited because employee 2 cannot sit beside employees 0, 1, and 3, simultaneously. Note that the company can also invite employees 1, 2, and 3, and give them their desired seats. The maximum number of employees that can be invited to the meeting is 3.

Example 2: Input: favorite = [1,2,0] Output: 3 Explanation: Each employee is the favorite person of at least one other employee, and the only way the company can invite them is if they invite every employee. The seating arrangement will be the same as that in the figure given in example 1: - Employee 0 will sit between employees 2 and 1. - Employee 1 will sit between employees 0 and 2. - Employee 2 will sit between employees 1 and 0. The maximum number of employees that can be invited to the meeting is 3.

Example 3:

Input: favorite = [3,0,1,4,1] Output: 4 Explanation: The above figure shows how the company will invite employees 0, 1, 3, and 4, and seat them at the round table. Employee 2 cannot be invited because the two spots next to their favorite employee 1 are taken. So the company leaves them out of the meeting. The maximum number of employees that can be invited to the meeting is 4.

Constraints:

n == favorite.length 2 <= n <= 105 0 <= favorite[i] <= n - 1 favorite[i] != i

==

Problem Number: 2040 URL: https://leetcode.com/problems/stamping-the-grid Title: 2132. Stamping the Grid Problem Description: You are given an m x n binary matrix grid where each cell is either 0 (empty) or 1 (occupied). You are then given stamps of size stampHeight x stampWidth. We want to fit the stamps such that they follow the given restrictions and requirements:

Cover all the empty cells. Do not cover any of the occupied cells. We can put as many stamps as we want. Stamps can overlap with each other. Stamps are not allowed to be rotated. Stamps must stay completely inside the grid.

Return true if it is possible to fit the stamps while following the given restrictions and requirements. Otherwise, return false. Example 1:

Input: grid = [[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0]], stampHeight = 4, stampWidth = 3 Output: true Explanation: We have two overlapping stamps (labeled 1 and 2 in the image) that are able to cover all the empty cells.

Example 2:

Input: grid = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]], stampHeight = 2, stampWidth = 2 Output: false Explanation: There is no way to fit the stamps onto all the empty cells without the stamps going outside the grid.

Constraints:

m == grid.length n == grid[r].length 1 <= m, n <= 105 1 <= m * n <= 2 * 105 grid[r][c] is either 0 or 1. 1 <= stampHeight, stampWidth <= 105

==

Problem Number: 2041 URL: https://leetcode.com/problems/earliest-possible-day-of-full-bloom Title: 2136. Earliest Possible Day of Full Bloom Problem Description: You have n flower seeds. Every seed must be planted first before it can begin to grow, then bloom. Planting a seed takes time and so does the growth of a seed. You are given two 0-indexed integer arrays plantTime and growTime, of length n each:

plantTime[i] is the number of full days it takes you to plant the ith seed. Every day, you can work on planting exactly one seed. You do not have to work on planting the same seed on consecutive days, but the planting of a seed is not complete until you have worked plantTime[i] days on planting it in total. growTime[i] is the number of full days it takes the ith seed to grow after being completely planted. After the last day of its growth, the flower blooms and stays bloomed forever.

From the beginning of day 0, you can plant the seeds in any order. Return the earliest possible day where all seeds are blooming. Example 1:

Input: plantTime = [1,4,3], growTime = [2,3,1] Output: 9 Explanation: The grayed out pots represent planting days, colored pots represent growing days, and the flower represents the day it blooms. One optimal way is: On day 0, plant the 0th seed. The seed grows for 2 full days and blooms on day 3. On days 1, 2, 3, and 4, plant the 1st seed. The seed grows for 3 full days and blooms on day 8. On days 5, 6, and 7, plant the 2nd seed. The seed grows for 1 full day and blooms on day 9. Thus, on day 9, all the seeds are blooming.

Example 2:

Input: plantTime = [1,2,3,2], growTime = [2,1,2,1] Output: 9 Explanation: The grayed out pots represent planting days, colored pots represent growing days, and the flower represents the day it blooms. One optimal way is: On day 1, plant the 0th seed. The seed grows for 2 full days and blooms on day 4. On days 0 and 3, plant the 1st seed. The seed grows for 1 full day and blooms on day 5. On days 2, 4, and 5, plant the 2nd seed. The seed grows for 2 full days and blooms on day 8. On days 6 and 7, plant the 3rd seed. The seed grows for 1 full day and blooms on day 9. Thus, on day 9, all the seeds are blooming.

Example 3: Input: plantTime = [1], growTime = [1] Output: 2 Explanation: On day 0, plant the 0th seed. The seed grows for 1 full day and blooms on day 2. Thus, on day 2, all the seeds are blooming.

Constraints:

n == plantTime.length == growTime.length 1 <= n <= 105 1 <= plantTime[i], growTime[i] <= 104

==

Problem Number: 2042 URL: https://leetcode.com/problems/maximum-running-time-of-n-computers Title: 2141. Maximum Running Time of N Computers Problem Description: You have n computers. You are given the integer n and a 0-indexed integer array batteries where the ith battery can run a computer for batteries[i] minutes. You are interested in running all n computers simultaneously using the given batteries. Initially, you can insert at most one battery into each computer. After that and at any integer time moment, you can remove a battery from a computer and insert another battery any number of times. The inserted battery can be a totally new battery or a battery from another computer. You may assume that the removing and inserting processes take no time. Note that the batteries cannot be recharged. Return the maximum number of minutes you can run all the n computers simultaneously.   Example 1:

Input: n = 2, batteries = [3,3,3] Output: 4 Explanation: Initially, insert battery 0 into the first computer and battery 1 into the second computer. After two minutes, remove battery 1 from the second computer and insert battery 2 instead. Note that battery 1 can still run for one minute. At the end of the third minute, battery 0 is drained, and you need to remove it from the first computer and insert battery 1 instead. By the end of the fourth minute, battery 1 is also drained, and the first computer is no longer running. We can run the two computers simultaneously for at most 4 minutes, so we return 4.

Example 2:

Input: n = 2, batteries = [1,1,1,1] Output: 2 Explanation: Initially, insert battery 0 into the first computer and battery 2 into the second computer. After one minute, battery 0 and battery 2 are drained so you need to remove them and insert battery 1 into the first computer and battery 3 into the second computer. After another minute, battery 1 and battery 3 are also drained so the first and second computers are no longer running. We can run the two computers simultaneously for at most 2 minutes, so we return 2.

Constraints:

1 <= n <= batteries.length <= 105 1 <= batteries[i] <= 109

==

Problem Number: 2043 URL: https://leetcode.com/problems/number-of-ways-to-divide-a-long-corridor Title: 2147. Number of Ways to Divide a Long Corridor Problem Description: Along a long library corridor, there is a line of seats and decorative plants. You are given a 0-indexed string corridor of length n consisting of letters 'S' and 'P' where each 'S' represents a seat and each 'P' represents a plant. One room divider has already been installed to the left of index 0, and another to the right of index n - 1. Additional room dividers can be installed. For each position between indices i - 1 and i (1 <= i <= n - 1), at most one divider can be installed. Divide the corridor into non-overlapping sections, where each section has exactly two seats with any number of plants. There may be multiple ways to perform the division. Two ways are different if there is a position with a room divider installed in the first way but not in the second way. Return the number of ways to divide the corridor. Since the answer may be very large, return it modulo 109 + 7. If there is no way, return 0.   Example 1:

Input: corridor = "SSPPSPS" Output: 3 Explanation: There are 3 different ways to divide the corridor. The black bars in the above image indicate the two room dividers already installed. Note that in each of the ways, each section has exactly two seats.

Example 2:

Input: corridor = ”PPSPSP” Output: 1 Explanation: There is only 1 way to divide the corridor, by not installing any additional dividers. Installing any would create some section that does not have exactly two seats.

Example 3:

Input: corridor = ”S” Output: 0 Explanation: There is no way to divide the corridor because there will always be a section that does not have exactly two seats.

Constraints:

n == corridor.length 1 <= n <= 105 corridor[i] is either ’S’ or ’P’.

==
Problem Number: 2044 URL: https://leetcode.com/problems/maximum-good-people-based-on-statements Title: 2151. Maximum Good People Based on Statements Problem Description: There are two types of persons:

The good person: The person who always tells the truth. The bad person: The person who might tell the truth and might lie.

You are given a 0-indexed 2D integer array statements of size n x n that represents the statements made by n people about each other. More specifically, statements[i][j] could be one of the following:

0 which represents a statement made by person i that person j is a bad person. 1 which represents a statement made by person i that person j is a good person. 2 represents that no statement is made by person i about person j.

Additionally, no person ever makes a statement about themselves. Formally, we have that statements[i][i] = 2 for all 0 <= i < n. Return the maximum number of people who can be good based on the statements made by the n people. Example 1:

Input: statements = [[2,1,2],[1,2,2],[2,0,2]] Output: 2 Explanation: Each person makes a single statement. - Person 0 states that person 1 is good. - Person 1 states that person 0 is good. - Person 2 states that person 1 is bad. Let’s take person 2 as the key. - Assuming that person 2 is a good person: - Based on the statement made by person 2, person 1 is a bad person. - Now we know for sure that person 1 is bad and person 2 is good. - Based on the statement made by person 1, and since person 1 is bad, they could be: - telling the truth. There will be a contradiction in this case and this assumption is invalid. - lying. In this case, person 0 is also a bad person and lied in their statement. - Following that person 2 is a good person, there will be only one good person in the group. - Assuming that person 2 is a bad person: - Based on the statement made by person 2, and since person 2 is bad, they could be: - telling the truth. Following this scenario, person 0 and 1 are both bad as explained before. - Following that person 2 is bad but told the truth, there will be no good persons in the group. - lying. In this case person 1 is a good person. - Since person 1 is a good person, person 0 is also a good person. - Following that person 2 is bad and lied, there will be two good persons in the group. We can see that at most 2 persons are good in the best case, so we return 2. Note that there is more than one way to arrive at this conclusion.

Example 2:

Input: statements = [[2,0],[0,2]] Output: 1 Explanation: Each person makes a single statement. - Person 0 states that person 1 is bad. - Person 1 states that person 0 is bad. Let's take person 0 as the key. - Assuming that person 0 is a good person: - Based on the statement made by person 0, person 1 is a bad person and was lying. - Following that person 0 is a good person, there will be only one good person in the group. - Assuming that person 0 is a bad person: - Based on the statement made by person 0, and since person 0 is bad, they could be: - telling the truth. Following this scenario, person 0 and 1 are both bad. - Following that person 0 is bad but told the truth, there will be no good persons in the group. - lying. In this case person 1 is a good person. - Following that person 0 is bad and lied, there will be only one good person in the group. We can see that at most, one person is good in the best case, so we return 1. Note that there is more than one way to arrive at this conclusion.

Constraints:

n == statements.length == statements[i].length 2 <= n <= 15 statements[i][j] is either 0, 1, or 2. statements[i][i] == 2

==

Problem Number: 2045 URL: https://leetcode.com/problems/find-substring-with-given-hash-value Title: 2156. Find Substring With Given Hash Value Problem Description: The hash of a 0-indexed string s of length k, given integers p and m, is computed using the following function:

hash(s, p, m) = (val(s[0]) * p0 + val(s[1]) * p1 + ... + val(s[k-1]) * pk-1) mod m.

Where val(s[i]) represents the index of s[i] in the alphabet from val('a') = 1 to val('z') = 26. You are given a string s and the integers power, modulo, k, and hashValue. Return sub, the first substring of s of length k such that hash(sub, power, modulo) == hashValue. The test cases will be generated such that an answer always exists. A substring is a contiguous non-empty sequence of characters within a string.   Example 1: Input: s = "leetcode", power = 7, modulo = 20, k = 2, hashValue = 0 Output: "ee" Explanation: The hash of "ee" can be computed to be hash("ee", 7, 20) = (5 * 1 + 5 * 7) mod 20 = 40 mod 20 = 0. "ee" is the first substring of length 2 with hashValue 0. Hence, we return "ee".

Example 2: Input: s = "fbxzaad", power = 31, modulo = 100, k = 3, hashValue = 32 Output: "fbx" Explanation: The hash of "fbx" can be computed to be hash("fbx", 31, 100) = (6 * 1 + 2 * 31 + 24 * 312) mod 100 = 23132 mod 100 = 32. The hash of "bxz" can be computed to be hash("bxz", 31, 100) = (2 * 1 + 24 * 31 + 26 * 312) mod 100 = 25732 mod 100 = 32. "fbx" is the first substring of length 3 with hashValue 32. Hence, we return "fbx". Note that "bxz" also has a hash of 32 but it appears later than "fbx".

Constraints:

1 <= k <= s.length <= 2 * 104 1 <= power, modulo <= 109 0 <= hashValue < modulo s consists of lowercase English letters only. The test cases are generated such that an answer always exists.

==

Problem Number: 2046 URL: https://leetcode.com/problems/groups-of-strings Title: 2157. Groups of Strings Problem Description: You are given a 0-indexed array of strings words. Each string consists of lowercase English letters only. No letter occurs more than once in any string of words. Two strings s1 and s2 are said to be connected if the set of letters of s2 can be obtained from the set of letters of s1 by any one of the following operations:

Adding exactly one letter to the set of the letters of s1. Deleting exactly one letter from the set of the letters of s1. Replacing exactly one letter from the set of the letters of s1 with any letter, including itself.

The array words can be divided into one or more non-intersecting groups. A string belongs to a group if any one of the following is true:

It is connected to at least one other string of the group. It is the only string present in the group.

Note that the strings in words should be grouped in such a manner that a string belonging to a group cannot be connected to a string present in any other group. It can be proved that such an arrangement is always unique. Return an array ans of size 2 where:

ans[0] is the maximum number of groups words can be divided into, and ans[1] is the size of the largest group.

Example 1: Input: words = ["a","b","ab","cde"] Output: [2,3] Explanation: - words[0] can be used to obtain words[1] (by replacing 'a' with 'b'), and words[2] (by adding 'b'). So words[0] is connected to words[1] and words[2]. - words[1] can be used to obtain words[0] (by replacing 'b' with 'a'), and words[2] (by adding 'a'). So words[1] is connected to words[0] and words[2]. - words[2] can be used to obtain words[0] (by deleting 'b'), and words[1] (by deleting 'a'). So words[2] is connected to words[0] and words[1]. - words[3] is not connected to any string in words. Thus, words can be divided into 2 groups ["a","b","ab"] and ["cde"]. The size of the largest group is 3.

Example 2: Input: words = ["a","ab","abc"] Output: [1,3] Explanation: - words[0] is connected to words[1]. - words[1] is connected to words[0] and words[2]. - words[2] is connected to words[1]. Since all strings are connected to each other, they should be grouped together. Thus, the size of the largest group is 3.

Constraints:

1 <= words.length <= 2 * 104 1 <= words[i].length <= 26 words[i] consists of lowercase English letters only. No letter occurs more than once in words[i].

==

Problem Number: 2047 URL: https://leetcode.com/problems/minimum-difference-in-sums-after-removal-of-elements Title: 2163. Minimum Difference in Sums After Removal of Elements Problem Description: You are given a 0-indexed integer array nums consisting of 3 * n elements. You are allowed to remove any subsequence of elements of size exactly n from nums. The remaining 2 * n elements will be divided into two equal parts:

The first n elements belonging to the first part and their sum is sumfirst. The next n elements belonging to the second part and their sum is sumsecond.

The difference in sums of the two parts is denoted as sumfirst - sumsecond.

For example, if sumfirst = 3 and sumsecond = 2, their difference is 1. Similarly, if sumfirst = 2 and sumsecond = 3, their difference is -1.

Return the minimum difference possible between the sums of the two parts after the removal of n elements. Example 1: Input: nums = [3,1,2] Output: -1 Explanation: Here, nums has 3 elements, so n = 1. Thus we have to remove 1 element from nums and divide the array into two equal parts. - If we remove nums[0] = 3, the array will be [1,2]. The difference in sums of the two parts will be 1 - 2 = -1. - If we remove nums[1] = 1, the array will be [3,2]. The difference in sums of the two parts will be 3 - 2 = 1. - If we remove nums[2] = 2, the array will be [3,1]. The difference in sums of the two parts will be 3 - 1 = 2. The minimum difference between sums of the two parts is min(-1,1,2) = -1.

Example 2: Input: nums = [7,9,5,8,1,3] Output: 1 Explanation: Here n = 2. So we must remove 2 elements and divide the remaining array into two parts containing two elements each. If we remove nums[2] = 5 and nums[3] = 8, the resultant array will be [7,9,1,3]. The difference in sums will be (7+9) - (1+3) = 12. To obtain the minimum difference, we should remove nums[1] = 9 and nums[4] = 1. The resultant array becomes [7,5,8,3]. The difference in sums of the two parts is (7+5) - (8+3) = 1. It can be shown that it is not possible to obtain a difference smaller than 1.

Constraints:

nums.length == 3 * n 1 <= n <= 105 1 <= nums[i] <= 105

==

Problem Number: 2048 URL: https://leetcode.com/problems/minimum-time-to-remove-all-cars-containing-illegal-goods Title: 2167. Minimum Time to Remove All Cars Containing Illegal Goods Problem Description: You are given a 0-indexed binary string s which represents a sequence of train cars. s[i] = '0' denotes that the ith car does not contain illegal goods and s[i] = '1' denotes that the ith car does contain illegal goods. As the train conductor, you would like to get rid of all the cars containing illegal goods. You can do any of the following three operations any number of times:

Remove a train car from the left end (i.e., remove s[0]) which takes 1 unit of time. Remove a train car from the right end (i.e., remove s[s.length - 1]) which takes 1 unit of time. Remove a train car from anywhere in the sequence which takes 2 units of time.

Return the minimum time to remove all the cars containing illegal goods. Note that an empty sequence of cars is considered to have no cars containing illegal goods. Example 1: Input: s = "1100101" Output: 5 Explanation: One way to remove all the cars containing illegal goods from the sequence is to - remove a car from the left end 2 times. Time taken is 2 * 1 = 2. - remove a car from the right end. Time taken is 1. - remove the car containing illegal goods found in the middle. Time taken is 2. This obtains a total time of 2 + 1 + 2 = 5.

An alternative way is to - remove a car from the left end 2 times. Time taken is 2 * 1 = 2. - remove a car from the right end 3 times. Time taken is 3 * 1 = 3. This also obtains a total time of 2 + 3 = 5.

5 is the minimum time taken to remove all the cars containing illegal goods. There are no other ways to remove them with less time.

Example 2: Input: s = "0010" Output: 2 Explanation: One way to remove all the cars containing illegal goods from the sequence is to - remove a car from the left end 3 times. Time taken is 3 * 1 = 3. This obtains a total time of 3.

Another way to remove all the cars containing illegal goods from the sequence is to - remove the car containing illegal goods found in the middle. Time taken is 2. This obtains a total time of 2.

Another way to remove all the cars containing illegal goods from the sequence is to - remove a car from the right end 2 times. Time taken is 2 * 1 = 2. This obtains a total time of 2.

2 is the minimum time taken to remove all the cars containing illegal goods. There are no other ways to remove them with less time. Constraints:

1 <= s.length <= 2 * 105 s[i] is either '0' or '1'.

==
Problem Number: 2049 URL: https://leetcode.com/problems/maximum-and-sum-of-array Title: 2172. Maximum AND Sum of Array Problem Description: You are given an integer array nums of length n and an integer numSlots such that 2 * numSlots >= n. There are numSlots slots numbered from 1 to numSlots. You have to place all n integers into the slots such that each slot contains at most two numbers. The AND sum of a given placement is the sum of the bitwise AND of every number with its respective slot number.

For example, the AND sum of placing the numbers [1, 3] into slot 1 and [4, 6] into slot 2 is equal to (1 AND 1) + (3 AND 1) + (4 AND 2) + (6 AND 2) = 1

+ 1 + 0 + 2 = 4.

Return the maximum possible AND sum of nums given numSlots slots. Example 1: Input: nums = [1,2,3,4,5,6], numSlots = 3 Output: 9 Explanation: One possible placement is [1, 4] into slot 1, [2, 6] into slot 2, and [3, 5] into slot 3. This gives the maximum AND sum of (1 AND 1) + (4 AND 1) + (2 AND 2) + (6 AND 2) + (3 AND 3) + (5 AND 3) = 1 + 0 + 2 + 2 + 3 + 1 = 9.

Example 2: Input: nums = [1,3,10,4,7,1], numSlots = 9 Output: 24 Explanation: One possible placement is [1, 1] into slot 1, [3] into slot 3, [4] into slot 4, [7] into slot 7, and [10] into slot 9. This gives the maximum AND sum of (1 AND 1) + (1 AND 1) + (3 AND 3) + (4 AND 4) + (7 AND 7) + (10 AND 9) = 1 + 1 + 3 + 4 + 7 + 8 = 24. Note that slots 2, 5, 6, and 8 are empty which is permitted.

Constraints:

n == nums.length 1 <= numSlots <= 9 1 <= n <= 2 * numSlots 1 <= nums[i] <= 15

==
Problem Number: 2050 URL: https://leetcode.com/problems/count-good-triplets-in-an-array Title: 2179. Count Good Triplets in an Array Problem Description: You are given two 0-indexed arrays nums1 and nums2 of length n, both of which are permutations of [0, 1, ..., n - 1]. A good triplet is a set of 3 distinct values which are present in increasing order by position both in nums1 and nums2. In other words, if we consider pos1v as the index of the value v in nums1 and pos2v as the index of the value v in nums2, then a good triplet will be a set (x, y, z) where 0 <= x, y, z <= n - 1, such that pos1x < pos1y < pos1z and pos2x < pos2y < pos2z. Return the total number of good triplets. Example 1: Input: nums1 = [2,0,1,3], nums2 = [0,1,2,3] Output: 1 Explanation: There are 4 triplets (x,y,z) such that pos1x < pos1y < pos1z. They are (2,0,1), (2,0,3), (2,1,3), and (0,1,3). Out of those triplets, only the triplet (0,1,3) satisfies pos2x < pos2y < pos2z. Hence, there is only 1 good triplet.

Example 2: Input: nums1 = [4,0,1,3,2], nums2 = [4,1,0,2,3] Output: 4 Explanation: The 4 good triplets are (4,0,3), (4,0,2), (4,1,3), and (4,1,2).

Constraints:

n == nums1.length == nums2.length 3 <= n <= 105 0 <= nums1[i], nums2[i] <= n - 1 nums1 and nums2 are permutations of [0, 1, ..., n - 1].

==
Problem Number: 2051 URL: https://leetcode.com/problems/count-array-pairs-divisible-by-k Title: 2183. Count Array Pairs Divisible by K Problem Description: Given a 0-indexed integer array nums of length n and an integer k, return the number of pairs (i, j) such that:

0 <= i < j <= n - 1 and nums[i] * nums[j] is divisible by k.

Example 1: Input: nums = [1,2,3,4,5], k = 2 Output: 7 Explanation: The 7 pairs of indices whose corresponding products are divisible by 2 are (0, 1), (0, 3), (1, 2), (1, 3), (1, 4), (2, 3), and (3, 4). Their products are 2, 4, 6, 8, 10, 12, and 20 respectively. Other pairs such as (0, 2) and (2, 4) have products 3 and 15 respectively, which are not divisible by 2.

Example 2: Input: nums = [1,2,3,4], k = 5 Output: 0 Explanation: There does not exist any pair of indices whose corresponding product is divisible by 5.

Constraints:

1 <= nums.length <= 105 1 <= nums[i], k <= 105

==
Problem Number: 2052 URL: https://leetcode.com/problems/minimum-time-to-finish-the-race Title: 2188. Minimum Time to Finish the Race Problem Description: You are given a 0-indexed 2D integer array tires where tires[i] = [fi, ri] indicates that the ith tire can finish its xth successive lap in fi * ri(x-1) seconds.

For example, if fi = 3 and ri = 2, then the tire would finish its 1st lap in 3 seconds, its 2nd lap in 3 * 2 = 6 seconds, its 3rd lap in 3 * 22 = 12 seconds, etc.

You are also given an integer changeTime and an integer numLaps. The race consists of numLaps laps and you may start the race with any tire. You have an unlimited supply of each tire and after every lap, you may change to any given tire (including the current tire type) if you wait changeTime seconds. Return the minimum time to finish the race. Example 1: Input: tires = [[2,3],[3,4]], changeTime = 5, numLaps = 4 Output: 21 Explanation: Lap 1: Start with tire 0 and finish the lap in 2 seconds. Lap 2: Continue with tire 0 and finish the lap in 2 * 3 = 6 seconds. Lap 3: Change tires to a new tire 0 for 5 seconds and then finish the lap in another 2 seconds. Lap 4: Continue with tire 0 and finish the lap in 2 * 3 = 6 seconds. Total time = 2 + 6 + 5 + 2 + 6 = 21 seconds. The minimum time to complete the race is 21 seconds.

Example 2: Input: tires = [[1,10],[2,2],[3,4]], changeTime = 6, numLaps = 5 Output: 25 Explanation: Lap 1: Start with tire 1 and finish the lap in 2 seconds. Lap 2: Continue with tire 1 and finish the lap in 2 * 2 = 4 seconds. Lap 3: Change tires to a new tire 1 for 6 seconds and then finish the lap in another 2 seconds. Lap 4: Continue with tire 1 and finish the lap in 2 * 2 = 4 seconds. Lap 5: Change tires to tire 0 for 6 seconds then finish the lap in another 1 second. Total time = 2 + 4 + 6 + 2 + 4 + 6 + 1 = 25 seconds. The minimum time to complete the race is 25 seconds.

Constraints:

1 <= tires.length <= 105 tires[i].length == 2 1 <= fi, changeTime <= 105 2 <= ri <= 105 1 <= numLaps <= 1000

==
Problem Number: 2053 URL: https://leetcode.com/problems/minimum-

number-of-moves-to-make-palindrome Title: 2193. Minimum Number of Moves to Make Palindrome Problem Description: You are given a string s consisting only of lowercase English letters. In one move, you can select any two adjacent characters of s and swap them. Return the minimum number of moves needed to make s a palindrome. Note that the input will be generated such that s can always be converted to a palindrome. Example 1: Input: s = "aabb" Output: 2 Explanation: We can obtain two palindromes from s, "abba" and "baab". - We can obtain "abba" from s in 2 moves: "aabb" -> "abab" -> "abba". - We can obtain "baab" from s in 2 moves: "aabb" -> "abab" -> "baab". Thus, the minimum number of moves needed to make s a palindrome is 2.

Example 2: Input: s = "letelt" Output: 2 Explanation: One of the palindromes we can obtain from s in 2 moves is "lettel". One of the ways we can obtain it is "letelt" -> "letetl" -> "lettel". Other palindromes such as "tleelt" can also be obtained in 2 moves. It can be shown that it is not possible to obtain a palindrome in less than 2 moves.

Constraints:

1 <= s.length <= 2000 s consists only of lowercase English letters. s can be converted to a palindrome using a finite number of moves.

==

Problem Number: 2054 URL: https://leetcode.com/problems/replace-non-coprime-numbers-in-array Title: 2197. Replace Non-Coprime Numbers in Array Problem Description: You are given an array of integers nums. Perform the following steps:

Find any two adjacent numbers in nums that are non-coprime. If no such numbers are found, stop the process. Otherwise, delete the two numbers and replace them with their LCM (Least Common Multiple). Repeat this process as long as you keep finding two adjacent non-coprime numbers.

Return the final modified array. It can be shown that replacing adjacent non-coprime numbers in any arbitrary order will lead to the same result. The test cases are generated such that the values in the final array are less than or equal to 108. Two values x and y are non-coprime if GCD(x, y) > 1 where GCD(x, y) is the Greatest Common Divisor of x and y. Example 1: Input: nums = [6,4,3,2,7,6,2] Output: [12,7,6] Explanation: - (6, 4) are non-coprime with LCM(6, 4) = 12. Now, nums = [12,3,2,7,6,2]. - (12, 3) are non-coprime with LCM(12, 3) = 12. Now, nums = [12,2,7,6,2]. - (12, 2) are non-coprime with LCM(12, 2) = 12. Now, nums = [12,7,6,2]. - (6, 2) are non-coprime with LCM(6, 2) = 6. Now, nums = [12,7,6]. There are no more adjacent non-coprime numbers in nums. Thus, the final modified array is [12,7,6]. Note that there are other ways to obtain the same resultant array.

Example 2: Input: nums = [2,2,1,1,3,3,3] Output: [2,1,1,3] Explanation: - (3, 3) are non-coprime with LCM(3, 3) = 3. Now, nums = [2,2,1,1,3,3]. - (3, 3) are non-coprime with LCM(3, 3) = 3. Now, nums = [2,2,1,1,3]. - (2, 2) are non-coprime with LCM(2, 2) = 2. Now, nums = [2,1,1,3]. There are no more adjacent non-coprime numbers in nums. Thus, the final modified array is [2,1,1,3]. Note that there are other ways to obtain the same resultant array.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105 The test cases are generated such that the values in the final array are less than or equal to 108.

== Problem Number: 2055 URL: https://leetcode.com/problems/minimum-weighted-subgraph-with-the-required-paths Title: 2203. Minimum Weighted Subgraph With the Required Paths Problem Description: You are given an integer n denoting the number of nodes of a weighted directed graph. The nodes are numbered from 0 to n - 1. You are also given a 2D integer array edges where edges[i] = [fromi, toi, weighti] denotes that there exists a directed edge from fromi to toi with weight weighti. Lastly, you are given three distinct integers src1, src2, and dest denoting three distinct nodes of the graph. Return the minimum weight of a subgraph of the graph such that it is possible to reach dest from both src1 and src2 via a set of edges of this subgraph. In case such a subgraph does not exist, return -1. A subgraph is a graph whose vertices and edges are subsets of the original graph. The weight of a subgraph is the sum of weights of its constituent edges. Example 1:

Input: n = 6, edges = [[0,2,2],[0,5,6],[1,0,3],[1,4,5],[2,1,1],[2,3,3],[2,3,4],[3,4,2],[4,5,1]], src1 = 0, src2 = 1, dest = 5 Output: 9 Explanation: The above figure represents the input graph. The blue edges represent one of the subgraphs that yield the optimal answer. Note that the subgraph [[1,0,3],[0,5,6]] also yields the optimal answer. It is not possible to get a subgraph with less weight satisfying all the constraints.

Example 2:

Input: n = 3, edges = [[0,1,1],[2,1,1]], src1 = 0, src2 = 1, dest = 2 Output: -1 Explanation: The above figure represents the input graph. It can be seen that there does not exist any path from node 1 to node 2, hence there are no subgraphs satisfying all the constraints.

Constraints:

3 <= n <= 105 0 <= edges.length <= 105 edges[i].length == 3 0 <= fromi, toi, src1, src2, dest <= n - 1 fromi != toi src1, src2, and dest are pairwise distinct. 1 <= weight[i] <= 105

== Problem Number: 2056 URL: https://leetcode.com/problems/minimum-white-tiles-after-covering-with-carpets Title: 2209. Minimum White Tiles After Covering With Carpets Problem Description: You are given a 0-indexed binary string floor, which represents the colors of tiles on a floor:

floor[i] = '0' denotes that the ith tile of the floor is colored black. On the other hand, floor[i] = '1' denotes that the ith tile of the floor is colored white.

You are also given numCarpets and carpetLen. You have numCarpets black carpets, each of length carpetLen tiles. Cover the tiles with the given carpets such that the number of white tiles still visible is minimum. Carpets may overlap one another. Return the minimum number of white tiles still visible. Example 1:

Input: floor = "10110101", numCarpets = 2, carpetLen = 2 Output: 2 Explanation: The figure above shows one way of covering the tiles with the carpets such that only 2 white tiles are visible. No other way of covering the tiles with the carpets can leave less than 2 white tiles visible.

Example 2:

Input: floor = "11111", numCarpets = 2, carpetLen = 3 Output: 0 Explanation: The figure above shows one way of covering the tiles with the carpets such that no white tiles are visible. Note that the carpets are able to overlap one another.

Constraints:

1 <= carpetLen <= floor.length <= 1000 floor[i] is either '0' or '1'. 1 <= numCarpets <= 1000

==

Problem Number: 2057 URL: https://leetcode.com/problems/longest-substring-of-one-repeating-character Title: 2213. Longest Substring of One Repeating Character Problem Description: You are given a 0-indexed string s. You are also given a 0-indexed string queryCharacters of length k and a 0-indexed array of integer indices queryIndices of length k, both of which are used to describe k queries. The ith query updates the character in s at index queryIndices[i] to the character queryCharacters[i]. Return an array lengths of length k where lengths[i] is the length of the longest substring of s consisting of only one repeating character after the ith query is performed. Example 1: Input: s = "babacc", queryCharacters = "bcb", queryIndices = [1,3,3] Output: [3,3,4] Explanation: - 1st query updates s = "bbbacc". The longest substring consisting of one repeating character is "bbb" with length 3. - 2nd query updates s = "bbbccc". The longest substring consisting of one repeating character can be "bbb" or "ccc" with length 3. - 3rd query updates s = "bbbbcc". The longest substring consisting of one repeating character is "bbbb" with length 4. Thus, we return [3,3,4].

Example 2: Input: s = "abyzz", queryCharacters = "aa", queryIndices = [2,1] Output: [2,3] Explanation: - 1st query updates s = "abazz". The longest substring consisting of one repeating character is "zz" with length 2. - 2nd query updates s = "aaazz". The longest substring consisting of one repeating character is "aaa" with length 3. Thus, we return [2,3].

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters. k == queryCharacters.length == queryIndices.length 1 <= k <= 105 queryCharacters consists of lowercase English letters. 0 <= queryIndices[i] < s.length

===

Problem Number: 2058 URL: https://leetcode.com/problems/maximum-value-of-k-coins-from-piles Title: 2218. Maximum Value of K Coins From Piles Problem Description: There are n piles of coins on a table. Each pile consists of a positive number of coins of assorted denominations. In one move, you can choose any coin on top of any pile, remove it, and add it to your wallet. Given a list piles, where piles[i] is a list of integers denoting the composition of the ith pile from top to bottom, and a positive integer k, return the maximum total value of coins you can have in your wallet if you choose exactly k coins optimally.   Example 1:

Input: piles = [[1,100,3],[7,8,9]], k = 2 Output: 101 Explanation: The above diagram shows the different ways we can choose k coins. The maximum total we can obtain is 101.

Example 2: Input: piles = [[100],[100],[100],[100],[100],[100],[1,1,1,1,1,1,700]], k = 7 Output: 706 Explanation: The maximum total can be obtained if we choose all coins from the last pile.

Constraints:

n == piles.length 1 <= n <= 1000 1 <= piles[i][j] <= 105 1 <= k <= sum(piles[i].length) <= 2000

===

Problem Number: 2059 URL: https://leetcode.com/problems/sum-of-scores-of-built-strings Title: 2223. Sum of Scores of Built Strings Problem Description: You are building a string s of length n one character at a time, prepending each new character to the front of the string. The strings are labeled from 1 to n, where the string with length i is labeled si.

For example, for s = ”abaca”, s1 == ”a”, s2 == ”ca”, s3 == ”aca”, etc.

The score of si is the length of the longest common prefix between si and sn (Note that s == sn). Given the final string s, return the sum of the score of every si.   Example 1: Input: s = ”babab” Output: 9 Explanation: For s1 == ”b”, the longest common prefix is ”b” which has a score of 1. For s2 == ”ab”, there is no common prefix so the score is 0. For s3 == ”bab”, the longest common prefix is ”bab” which has a score of 3. For s4 == ”abab”, there is no common prefix so the score is 0. For s5 == ”babab”, the longest common prefix is ”babab” which has a score of 5. The sum of the scores is 1 + 0 + 3 + 0 + 5 = 9, so we return 9. Example 2: Input: s = ”azbazbzaz” Output: 14 Explanation: For s2 == ”az”, the longest common prefix is ”az” which has a score of 2. For s6 == ”azbzaz”, the longest common prefix is ”azb” which has a score of 3. For s9 == ”azbazbzaz”, the longest common prefix is ”azbazbzaz” which has a score of 9. For all other si, the score is 0. The sum of the scores is 2 + 3 + 9 = 14, so we return 14.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters.

==
Problem Number: 2060 URL: https://leetcode.com/problems/encrypt-and-decrypt-strings Title: 2227. Encrypt and Decrypt Strings Problem Description: You are given a character array keys containing unique characters and a string array values containing strings of length 2. You are also given another string array dictionary that contains all permitted original strings after decryption. You should implement a data structure that can encrypt or decrypt a 0-indexed string. A string is encrypted with the following process:

For each character c in the string, we find the index i satisfying keys[i] == c in keys. Replace c with values[i] in the string.

Note that in case a character of the string is not present in keys, the encryption process cannot be carried out, and an empty string "" is returned. A string is decrypted with the following process:

For each substring s of length 2 occurring at an even index in the string, we find an i such that values[i] == s. If there are multiple valid i, we choose any one of them. This means a string could have multiple possible strings it can decrypt to. Replace s with keys[i] in the string.

Implement the Encrypter class:

Encrypter(char[] keys, String[] values, String[] dictionary) Initializes the Encrypter class with keys, values, and dictionary. String encrypt(String word1) Encrypts word1 with the encryption process described above and returns the encrypted string. int decrypt(String word2) Returns the number of possible strings word2 could decrypt to that also appear in dictionary.

Example 1: Input ["Encrypter", "encrypt", "decrypt"] [[['a', 'b', 'c', 'd'], ["ei", "zf", "ei", "am"], ["abcd", "acbd", "adbc", "badc", "dacb", "cadb", "cbda", "abad"]], ["abcd"], ["eizfeiam"]] Output [null, "eizfeiam", 2]

Explanation Encrypter encrypter = new Encrypter([['a', 'b', 'c', 'd'], ["ei", "zf", "ei", "am"], ["abcd", "acbd", "adbc", "badc", "dacb", "cadb", "cbda", "abad"]); encrypter.encrypt("abcd"); // return "eizfeiam". // 'a' maps to "ei", 'b' maps to "zf", 'c' maps to "ei", and 'd' maps to "am". encrypter.decrypt("eizfeiam"); // return 2. // "ei" can map to 'a' or 'c', "zf" maps to 'b', and "am" maps to 'd'. // Thus, the possible strings after decryption are "abad", "cbad", "abcd", and "cbcd". // 2 of those strings, "abad" and "abcd", appear in dictionary, so the answer is 2.

Constraints:

1 <= keys.length == values.length <= 26 values[i].length == 2 1 <= dictionary.length <= 100 1 <= dictionary[i].length <= 100 All keys[i] and dictionary[i] are unique. 1 <= word1.length <= 2000 1 <= word2.length <= 200 All word1[i] appear in keys. word2.length is even. keys, values[i], dictionary[i], word1, and word2 only contain lowercase English letters. At most 200 calls will be made to encrypt and decrypt in total.

==

Problem Number: 2061 URL: https://leetcode.com/problems/maximum-total-beauty-of-the-gardens Title: 2234. Maximum Total Beauty of the Gardens Problem Description: Alice is a caretaker of n gardens and she wants to plant flowers to maximize the total beauty of all her gardens. You are given a 0-indexed integer array flowers of size n, where flowers[i] is the number of flowers already planted in the ith garden. Flowers that are already planted cannot be removed. You are then given another integer newFlowers, which is the maximum number of flowers that Alice can additionally plant. You are also given the integers target, full, and partial. A garden is considered complete if it has at least target flowers. The total beauty of the gardens is then determined as the sum of the following:

The number of complete gardens multiplied by full. The minimum number of flowers in any of the incomplete gardens multiplied by partial. If there are no incomplete gardens, then this value will be 0.

Return the maximum total beauty that Alice can obtain after planting at most newFlowers flowers. Example 1: Input: flowers = [1,3,1,1], newFlowers = 7, target = 6, full = 12, partial = 1 Output: 14 Explanation: Alice can plant - 2 flowers in the 0th garden - 3 flowers in the 1st garden - 1 flower in the 2nd garden - 1 flower in the 3rd garden The gardens will then be [3,6,2,2]. She planted a total of 2 + 3 + 1 + 1 = 7 flowers. There is 1 garden that is complete. The minimum number of flowers in the incomplete gardens is 2. Thus, the total beauty is 1 * 12 + 2 * 1 = 12 + 2 = 14. No other way of planting flowers can obtain a total beauty higher than 14.

Example 2: Input: flowers = [2,4,5,3], newFlowers = 10, target = 5, full = 2, partial = 6 Output: 30 Explanation: Alice can plant - 3 flowers in the 0th garden - 0 flowers in the 1st garden - 0 flowers in the 2nd garden - 2 flowers in the 3rd garden The gardens will then be [5,4,5,5]. She planted a total of 3 + 0 + 0 + 2 = 5 flowers. There are 3 gardens that are complete. The minimum number of flowers in the incomplete gardens is 4. Thus, the total beauty is 3 * 2 + 4 * 6 = 6 + 24 = 30. No other way of planting flowers can obtain a total beauty higher than 30. Note that Alice could make all the gardens complete but in this case, she would obtain a lower total beauty.

Constraints:

1 <= flowers.length <= 105 1 <= flowers[i], target <= 105 1 <= newFlowers <= 1010 1 <= full, partial <= 105

== Problem Number: 2062 URL: https://leetcode.com/problems/maximum-score-of-a-node-sequence Title: 2242. Maximum Score of a Node Sequence Problem Description: There is an undirected graph with n nodes, numbered from 0 to n - 1. You are given a 0-indexed integer array scores of length n where scores[i] denotes the score of node i. You are also given a 2D integer array edges where edges[i] = [ai, bi] denotes that there exists an undirected edge connecting nodes ai and bi. A node sequence is valid if it meets the following conditions:

There is an edge connecting every pair of adjacent nodes in the sequence. No node appears more than once in the sequence.

The score of a node sequence is defined as the sum of the scores of the nodes in the sequence. Return the maximum score of a valid node sequence with a length of 4. If no such sequence exists, return -1.   Example 1:

Input: scores = [5,2,9,8,4], edges = [[0,1],[1,2],[2,3],[0,2],[1,3],[2,4]] Output: 24 Explanation: The figure above shows the graph and the chosen node sequence [0,1,2,3]. The score of the node sequence is 5 + 2 + 9 + 8 = 24. It can be shown that no other node sequence has a score of more than 24. Note that the sequences [3,1,2,0] and [1,0,2,3] are also valid and have a score of 24. The sequence [0,3,2,4] is not valid since no edge connects nodes 0 and 3.

Example 2:

Input: scores = [9,20,6,4,11,12], edges = [[0,3],[5,3],[2,4],[1,3]] Output: -1 Explanation: The figure above shows the graph. There are no valid node sequences of length 4, so we return -1.

Constraints:

n == scores.length 4 <= n <= 5 * 104 1 <= scores[i] <= 108 0 <= edges.length <= 5 * 104 edges[i].length == 2 0 <= ai, bi <= n - 1 ai != bi There are no duplicate edges.

== Problem Number: 2063 URL: https://leetcode.com/problems/longest-path-with-different-adjacent-characters Title: 2246. Longest Path With Different Adjacent Characters Problem Description: You are given a tree (i.e. a connected, undirected graph that has no cycles) rooted at node 0 consisting of n nodes numbered from 0 to n - 1. The tree is represented by a 0-indexed array parent of size n, where parent[i] is the parent of node i. Since node 0 is the root, parent[0] == -1. You are also given a string s of length n, where s[i] is the character assigned to node i. Return the length of the longest path in the tree such that no pair of adjacent nodes on the path have the same character assigned to them.   Example 1:

Input: parent = [-1,0,0,1,1,2], s = "abacbe" Output: 3 Explanation: The longest path where each two adjacent nodes have different characters in the tree is the path: 0 -> 1 -> 3. The length of this path is 3, so 3 is returned. It can be proven that there is no longer path that satisfies the conditions.

Example 2:

Input: parent = [-1,0,0,0], s = "aabc" Output: 3 Explanation: The longest path where each two adjacent nodes have different characters is the path: 2 -> 0 -> 3. The length of this path is 3, so 3 is returned.

Constraints:

n == parent.length == s.length 1 <= n <= 105 0 <= parent[i] <= n - 1 for all i >= 1 parent[0] == -1 parent represents a valid tree. s consists of only lowercase English letters.

==
Problem Number: 2064 URL: https://leetcode.com/problems/number-of-flowers-in-full-bloom Title: 2251. Number of Flowers in Full Bloom Problem Description: You are given a 0-indexed 2D integer array flowers, where flowers[i] = [starti, endi] means the ith flower will be in full bloom from starti to endi (inclusive). You are also given a 0-indexed integer array people of size n, where people[i] is the time that the ith person will arrive to see the flowers. Return an integer array answer of size n, where answer[i] is the number of flowers that are in full bloom when the ith person arrives. Example 1:

Input: flowers = [[1,6],[3,7],[9,12],[4,13]], poeple = [2,3,7,11] Output: [1,2,2,2] Explanation: The figure above shows the times when the flowers are in full bloom and when the people arrive. For each person, we return the number of flowers in full bloom during their arrival.

Example 2:

Input: flowers = [[1,10],[3,3]], poeple = [3,3,2] Output: [2,2,1] Explanation: The figure above shows the times when the flowers are in full bloom and when the people arrive. For each person, we return the number of flowers in full bloom during their arrival.

Constraints:

1 <= flowers.length <= 5 * 104 flowers[i].length == 2 1 <= starti <= endi <= 109 1 <= people.length <= 5 * 104 1 <= people[i] <= 109

==
Problem Number: 2065 URL: https://leetcode.com/problems/escape-the-spreading-fire Title: 2258. Escape the Spreading Fire Problem Description: You are given a 0-indexed 2D integer array grid of size m x n which represents a field. Each cell has one of three values:

0 represents grass, 1 represents fire, 2 represents a wall that you and fire cannot pass through.

You are situated in the top-left cell, (0, 0), and you want to travel to the safehouse at the bottom-right cell, (m - 1, n - 1). Every minute, you may move to an adjacent grass cell. After your move, every fire cell will spread to all adjacent cells that are not walls. Return the maximum number of minutes that you can stay in your initial position before moving while still safely reaching the safehouse. If this is impossible, return -1. If you can always reach the safehouse regardless of the minutes stayed, return 109. Note that even if the fire spreads to the safehouse immediately after you have reached it, it will be counted as safely reaching the safehouse. A cell is adjacent to another cell if the former is directly north, east, south, or west of the latter (i.e., their sides are touching). Example 1:

Input: grid = [[0,2,0,0,0,0,0],[0,0,0,2,2,1,0],[0,2,0,0,1,2,0],[0,0,2,2,2,0,2],[0,0,0,0,0,0,0]] Output: 3 Explanation: The figure above shows the scenario where you stay in the initial position for 3 minutes. You will still be able to safely reach the safehouse. Staying for more than 3 minutes will not allow you to safely reach the safehouse. Example 2:

Input: grid = [[0,0,0,0],[0,1,2,0],[0,2,0,0]] Output: -1 Explanation: The figure above shows the scenario where you immediately move towards the safehouse. Fire will spread to any cell you move towards and it is impossible to safely reach the safehouse. Thus, -1 is returned.

Example 3:

Input: grid = [[0,0,0],[2,2,0],[1,2,0]] Output: 1000000000 Explanation: The figure above shows the initial grid. Notice that the fire is contained by walls and you will always be able to safely reach the safehouse. Thus, 109 is returned.

Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 300 4 <= m * n <= 2 * 104 grid[i][j] is either 0, 1, or 2. grid[0][0] == grid[m - 1][n - 1] == 0

==
Problem Number: 2066 URL: https://leetcode.com/problems/total-appeal-of-a-string Title: 2262. Total Appeal of A String Problem Description: The appeal of a string is the number of distinct characters found in the string.

For example, the appeal of "abbca" is 3 because it has 3 distinct characters: 'a', 'b', and 'c'.

Given a string s, return the total appeal of all of its substrings. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "abbca" Output: 28 Explanation: The following are the substrings of "abbca": - Substrings of length 1: "a", "b", "b", "c", "a" have an appeal of 1, 1, 1, 1, and 1 respectively. The sum is 5. - Substrings of length 2: "ab", "bb", "bc", "ca" have an appeal of 2, 1, 2, and 2 respectively. The sum is 7. - Substrings of length 3: "abb", "bbc", "bca" have an appeal of 2, 2, and 3 respectively. The sum is 7. - Substrings of length 4: "abbc", "bbca" have an appeal of 3 and 3 respectively. The sum is 6. - Substrings of length 5: "abbca" has an appeal of 3. The sum is 3. The total sum is 5 + 7 + 7 + 6 + 3 = 28.

Example 2: Input: s = "code" Output: 20 Explanation: The following are the substrings of "code": - Substrings of length 1: "c", "o", "d", "e" have an appeal of 1, 1, 1, and 1 respectively. The sum is 4. - Substrings of length 2: "co", "od", "de" have an appeal of 2, 2, and 2 respectively. The sum is 6. - Substrings of length 3: "cod", "ode" have an appeal of 3 and 3 respectively. The sum is 6. - Substrings of length 4: "code" has an appeal of 4. The sum is 4. The total sum is 4 + 6 + 6 + 4 = 20.

Constraints:

1 <= s.length <= 105 s consists of lowercase English letters.

==

Problem Number: 2067 URL: https://leetcode.com/problems/check-if-there-is-a-valid-parentheses-string-path Title: 2267. Check if There Is a Valid Parentheses String Path Problem Description: A parentheses string is a non-empty string consisting only of '(' and ')'. It is valid if any of the following conditions is true:

It is (). It can be written as AB (A concatenated with B), where A and B are valid parentheses strings. It can be written as (A), where A is a valid parentheses string.

You are given an m x n matrix of parentheses grid. A valid parentheses string path in the grid is a path satisfying all of the following conditions:

The path starts from the upper left cell (0, 0). The path ends at the bottom-right cell (m - 1, n - 1). The path only ever moves down or right. The resulting parentheses string formed by the path is valid.

Return true if there exists a valid parentheses string path in the grid. Otherwise, return false.   Example 1:

Input: grid = [["(","(","("],[")","(",")"],["(","(",")"],["(","(",")"]] Output: true Explanation: The above diagram shows two possible paths that form valid parentheses strings. The first path shown results in the valid parentheses string "()(())". The second path shown results in the valid parentheses string "((()))". Note that there may be other valid parentheses string paths.

Example 2:

Input: grid = [[")",")"],["(","("]] Output: false Explanation: The two possible paths form the parentheses strings "))(" and ")((". Since neither of them are valid parentheses strings, we return false.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 100 grid[i][j] is either '(' or ')'.

==

Problem Number: 2068 URL: https://leetcode.com/problems/substring-with-largest-variance Title: 2272. Substring With Largest Variance Problem Description: The variance of a string is defined as the largest difference between the number of occurrences of any 2 characters present in the string. Note the two characters may or may not be the same. Given a string s consisting of lowercase English letters only, return the largest variance possible among all substrings of s. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "aababbb" Output: 3 Explanation: All possible variances along with their respective substrings are listed below: - Variance 0 for substrings "a", "aa", "ab", "abab", "aababb", "ba", "b", "bb", and "bbb". - Variance 1 for substrings "aab", "aba", "abb", "aabab", "ababb", "aababbb", and "bab". - Variance 2 for substrings "aaba", "ababbb", "abbb", and "babb". - Variance 3 for substring "babbb". Since the largest possible variance is 3, we return it.

Example 2: Input: s = "abcde" Output: 0 Explanation: No letter occurs more than once in s, so the variance of every substring is 0.

Constraints:

1 <= s.length <= 104 s consists of lowercase English letters.

==

Problem Number: 2069 URL: https://leetcode.com/problems/count-integers-in-intervals Title: 2276. Count Integers in Intervals Problem Description: Given an empty set of intervals, implement a data structure that can:

Add an interval to the set of intervals. Count the number of integers that are present in at least one interval.

Implement the CountIntervals class:

CountIntervals() Initializes the object with an empty set of intervals. void add(int left, int right) Adds the interval [left, right] to the set of intervals. int count() Returns the number of integers that are present in at least one interval.

Note that an interval [left, right] denotes all the integers x where left <= x <= right. Example 1: Input ["CountIntervals", "add", "add", "count", "add", "count"] [[], [2, 3], [7, 10], [], [5, 8], []] Output [null, null, null, 6, null, 8]

Explanation CountIntervals countIntervals = new CountIntervals(); // initialize the object with an empty set of intervals. countIntervals.add(2, 3); // add [2, 3] to the set of intervals. countIntervals.add(7, 10); // add [7, 10] to the set of intervals. countIntervals.count(); // return 6 // the integers 2 and 3 are present in the interval [2, 3]. // the integers 7, 8, 9, and 10 are present in the interval [7, 10]. countIntervals.add(5, 8); // add [5, 8] to the set of intervals. countIntervals.count(); // return 8 // the integers 2 and 3 are present in the interval [2, 3]. // the integers 5 and 6 are present in the interval [5, 8]. // the integers 7 and 8 are present in the intervals [5, 8] and [7, 10]. // the integers 9 and 10 are present in the interval [7, 10].

Constraints:

1 <= left <= right <= 109 At most 105 calls in total will be made to add and count. At least one call will be made to count.

==

Problem Number: 2070 URL: https://leetcode.com/problems/sum-of-total-strength-of-wizards Title: 2281. Sum of Total Strength of Wizards Problem Description: As the ruler of a kingdom, you have an army of wizards at your command. You are given a 0-indexed integer array strength, where strength[i] denotes the strength of the ith wizard. For a contiguous group of wizards (i.e. the wizards' strengths form a subarray of strength), the total strength is defined as the product of the following two values:

The strength of the weakest wizard in the group. The total of all the individual strengths of the wizards in the group.

Return the sum of the total strengths of all contiguous groups of wizards. Since the answer may be very large, return it modulo 109 + 7. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: strength = [1,3,1,2] Output: 44 Explanation: The following are all the contiguous groups of wizards: - [1] from [1,3,1,2] has a total strength of min([1]) * sum([1]) = 1 * 1 = 1 - [3] from [1,3,1,2] has a total strength of min([3]) * sum([3]) = 3 * 3 = 9 - [1] from [1,3,1,2] has a total strength of min([1]) * sum([1]) = 1 * 1 = 1 - [2] from [1,3,1,2] has a total strength of min([2]) * sum([2]) = 2 * 2 = 4 - [1,3] from [1,3,1,2] has a total strength of min([1,3]) * sum([1,3]) = 1 * 4 = 4 - [3,1] from [1,3,1,2] has a total strength of min([3,1]) * sum([3,1]) = 1 * 4 = 4 - [1,2] from [1,3,1,2] has a total strength of min([1,2]) * sum([1,2]) = 1 * 3 = 3 - [1,3,1] from [1,3,1,2] has a total strength of min([1,3,1]) * sum([1,3,1]) = 1 * 5 = 5 - [3,1,2] from [1,3,1,2] has a total strength of min([3,1,2]) * sum([3,1,2]) = 1 * 6 = 6 - [1,3,1,2] from [1,3,1,2] has a total strength of min([1,3,1,2]) * sum([1,3,1,2]) = 1 * 7 = 7 The sum of all the total strengths is 1 + 9 + 1 + 4 + 4 + 4 + 3 + 5 + 6 + 7 = 44.

Example 2: Input: strength = [5,4,6] Output: 213 Explanation: The following are all the contiguous groups of wizards: - [5] from [5,4,6] has a total strength of min([5]) * sum([5]) = 5 * 5 = 25 - [4] from [5,4,6] has a total strength of min([4]) * sum([4]) = 4 * 4 = 16 - [6] from [5,4,6] has a total strength of min([6]) * sum([6]) = 6 * 6 = 36 - [5,4] from [5,4,6] has a total strength of min([5,4]) * sum([5,4]) = 4 * 9 = 36 - [4,6] from [5,4,6] has a total strength of min([4,6]) * sum([4,6]) = 4 * 10 = 40 - [5,4,6] from [5,4,6] has a total strength of min([5,4,6]) * sum([5,4,6]) = 4 * 15 = 60 The sum of all the total strengths is 25 + 16 + 36 + 36 + 40 + 60 = 213.

Constraints:

1 <= strength.length <= 105 1 <= strength[i] <= 109

==

Problem Number: 2071 URL: https://leetcode.com/problems/booking-concert-tickets-in-groups Title: 2286. Booking Concert Tickets in Groups Problem Description: A concert hall has n rows numbered from 0 to n - 1, each with m seats, numbered from 0 to m - 1. You need to design a ticketing system that can allocate seats in the following cases:

If a group of k spectators can sit together in a row. If every member of a group of k spectators can get a seat. They may or may not sit together.

Note that the spectators are very picky. Hence:

They will book seats only if each member of their group can get a seat with row number less than or equal to maxRow. maxRow can vary from group to group. In case there are multiple rows to choose from, the row with the smallest number is chosen. If there are multiple seats to choose in the same row, the seat with the smallest number is chosen.

Implement the BookMyShow class:

BookMyShow(int n, int m) Initializes the object with n as number of rows and m as number of seats per row. int[] gather(int k, int maxRow) Returns an array of length 2 denoting the row and seat number (respectively) of the first seat being allocated to the k members of the group, who must sit together. In other words, it returns the smallest possible r and c such that all [c, c + k - 1] seats are valid and empty in row r, and r <= maxRow. Returns [] in case it is not possible to allocate seats to the group. boolean scatter(int k, int maxRow) Returns true if all k members of the group can be allocated seats in rows 0 to maxRow, who may or may not sit together. If the seats can be allocated, it allocates k seats to the group with the smallest row numbers, and the smallest possible seat numbers in each row. Otherwise, returns false.

Example 1: Input ["BookMyShow", "gather", "gather", "scatter", "scatter"] [[2, 5], [4, 0], [2, 0], [5, 1], [5, 1]] Output [null, [0, 0], [], true, false]

Explanation BookMyShow bms = new BookMyShow(2, 5); // There are 2 rows with 5 seats each bms.gather(4, 0); // return [0, 0] // The group books seats [0, 3] of row 0. bms.gather(2, 0); // return [] // There is only 1 seat left in row 0, // so it is not possible to book 2 consecutive seats. bms.scatter(5, 1); // return True // The group books seat 4 of row 0 and seats [0, 3] of row 1. bms.scatter(5, 1); // return False // There is only one seat left in the hall.

Constraints:

1 <= n <= 5 * 104 1 <= m, k <= 109 0 <= maxRow <= n - 1 At most 5 * 104 calls in total will be made to gather and scatter.

==

Problem Number: 2072 URL: https://leetcode.com/problems/minimum-obstacle-removal-to-reach-corner Title: 2290. Minimum Obstacle Removal to

Reach Corner Problem Description: You are given a 0-indexed 2D integer array grid of size m x n. Each cell has one of two values:

0 represents an empty cell, 1 represents an obstacle that may be removed.

You can move up, down, left, or right from and to an empty cell. Return the minimum number of obstacles to remove so you can move from the upper left corner (0, 0) to the lower right corner (m - 1, n - 1).   Example 1:

Input: grid = [[0,1,1],[1,1,0],[1,1,0]] Output: 2 Explanation: We can remove the obstacles at (0, 1) and (0, 2) to create a path from (0, 0) to (2, 2). It can be shown that we need to remove at least 2 obstacles, so we return 2. Note that there may be other ways to remove 2 obstacles to create a path.

Example 2:

Input: grid = [[0,1,0,0,0],[0,1,0,1,0],[0,0,0,1,0]] Output: 0 Explanation: We can move from (0, 0) to (2, 4) without removing any obstacles, so we return 0.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 2 <= m * n <= 105 grid[i][j] is either 0 or 1. grid[0][0] == grid[m - 1][n - 1] == 0

==
Problem Number: 2073 URL: https://leetcode.com/problems/design-a-text-editor Title: 2296. Design a Text Editor Problem Description: Design a text editor with a cursor that can do the following:

Add text to where the cursor is. Delete text from where the cursor is (simulating the backspace key). Move the cursor either left or right.

When deleting text, only characters to the left of the cursor will be deleted. The cursor will also remain within the actual text and cannot be moved beyond it. More formally, we have that 0 <= cursor.position <= currentText.length always holds. Implement the TextEditor class:

TextEditor() Initializes the object with empty text. void addText(string text) Appends text to where the cursor is. The cursor ends to the right of text. int deleteText(int k) Deletes k characters to the left of the cursor. Returns the number of characters actually deleted. string cursorLeft(int k) Moves the cursor to the left k times. Returns the last min(10, len) characters to the left of the cursor, where len is the number of characters to the left of the cursor. string cursorRight(int k) Moves the cursor to the right k times. Returns the last min(10, len) characters to the left of the cursor, where len is the number of characters to the left of the cursor.

Example 1: Input ["TextEditor", "addText", "deleteText", "addText", "cursorRight", "cursorLeft", "deleteText", "cursorLeft", "cursorRight"] [[], ["leetcode"], [4], ["practice"], [3], [8], [10], [2], [6]] Output [null, null, 4, null, "etpractice", "leet", 4, "", "practi"]

Explanation TextEditor textEditor = new TextEditor(); // The current text is ”|”. (The ’|’ character represents the cursor) textEditor.addText(”leetcode”); // The current text is ”leetcode|”. textEditor.deleteText(4); // return 4 // The current text is ”leet|”. // 4 characters were deleted. textEditor.addText(”practice”); // The current text is ”leetpractice|”. textEditor.cursorRight(3); // return ”etpractice” // The current text is ”leetpractice|”. // The cursor cannot be moved beyond the actual text and thus did not move. // ”etpractice” is the last 10 characters to the left of the cursor. textEditor.cursorLeft(8); // return ”leet” // The current text is ”leet|practice”. // ”leet” is the last min(10, 4) = 4 characters to the left of the cursor. textEditor.deleteText(10); // return 4 // The current text is ”|practice”. // Only 4 characters were deleted. textEditor.cursorLeft(2); // return ”” // The current text is ”|practice”. // The cursor cannot be moved beyond the actual text and thus did not move. // ”” is the last min(10, 0) = 0 characters to the left of the cursor. textEditor.cursorRight(6); // return ”practi” // The current text is ”practi|ce”. // ”practi” is the last min(10, 6) = 6 characters to the left of the cursor.

Constraints:

1 <= text.length, k <= 40 text consists of lowercase English letters. At most 2 * 104 calls in total will be made to addText, deleteText, cursorLeft and cursorRight.

Follow-up: Could you find a solution with time complexity of O(k) per call?

==

Problem Number: 2074 URL: https://leetcode.com/problems/match-substring-after-replacement Title: 2301. Match Substring After Replacement Problem Description: You are given two strings s and sub. You are also given a 2D character array mappings where mappings[i] = [oldi, newi] indicates that you may perform the following operation any number of times:

Replace a character oldi of sub with newi.

Each character in sub cannot be replaced more than once. Return true if it is possible to make sub a substring of s by replacing zero or more characters according to mappings. Otherwise, return false. A substring is a contiguous non-empty sequence of characters within a string. Example 1: Input: s = ”fool3e7bar”, sub = ”leet”, mappings = [[”e”,”3”],[”t”,”7”],[”t”,”8”]] Output: true Explanation: Replace the first ’e’ in sub with ’3’ and ’t’ in sub with ’7’. Now sub = ”l3e7” is a substring of s, so we return true. Example 2: Input: s = ”fooleetbar”, sub = ”f00l”, mappings = [[”o”,”0”]] Output: false Explanation: The string ”f00l” is not a substring of s and no replacements can be made. Note that we cannot replace ’0’ with ’o’.

Example 3: Input: s = ”Fool33tbaR”, sub = ”leetd”, mappings = [[”e”,”3”],[”t”,”7”],[”t”,”8”],[”d”,”b”],[”p”,”b”]] Output: true Explanation: Replace the first and second ’e’ in sub with ’3’ and ’d’ in sub with ’b’. Now sub = ”l33tb” is a substring of s, so we return true.

Constraints:

1 <= sub.length <= s.length <= 5000 0 <= mappings.length <= 1000 mappings[i].length == 2 oldi != newi s and sub consist of uppercase and lowercase English letters and digits. oldi and newi are either uppercase or lowercase English letters or digits.

==
Problem Number: 2075 URL: https://leetcode.com/problems/count-subarrays-with-score-less-than-k Title: 2302. Count Subarrays With Score Less Than K Problem Description: The score of an array is defined as the product of its sum and its length.

For example, the score of [1, 2, 3, 4, 5] is (1 + 2 + 3 + 4 + 5) * 5 = 75.

Given a positive integer array nums and an integer k, return the number of non-empty subarrays of nums whose score is strictly less than k. A subarray is a contiguous sequence of elements within an array.   Example 1: Input: nums = [2,1,4,3,5], k = 10 Output: 6 Explanation: The 6 subarrays having scores less than 10 are: - [2] with score 2 * 1 = 2. - [1] with score 1 * 1 = 1. - [4] with score 4 * 1 = 4. - [3] with score 3 * 1 = 3. - [5] with score 5 * 1 = 5. - [2,1] with score (2 + 1) * 2 = 6. Note that subarrays such as [1,4] and [4,3,5] are not considered because their scores are 10 and 36 respectively, while we need scores strictly less than 10. Example 2: Input: nums = [1,1,1], k = 5 Output: 5 Explanation: Every subarray except [1,1,1] has a score less than 5. [1,1,1] has a score (1 + 1 + 1) * 3 = 9, which is greater than 5. Thus, there are 5 subarrays having scores less than 5.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105 1 <= k <= 1015

==
Problem Number: 2076 URL: https://leetcode.com/problems/naming-a-company Title: 2306. Naming a Company Problem Description: You are given an array of strings ideas that represents a list of names to be used in the process of naming a company. The process of naming a company is as follows:

Choose 2 distinct names from ideas, call them ideaA and ideaB. Swap the first letters of ideaA and ideaB with each other. If both of the new names are not found in the original ideas, then the name ideaA ideaB (the concatenation of ideaA and ideaB, separated by a space) is a valid company name. Otherwise, it is not a valid name.

Return the number of distinct valid names for the company.   Example 1: Input: ideas = ["coffee","donuts","time","toffee"] Output: 6 Explanation: The following selections are valid: - ("coffee", "donuts"): The company name created is "doffee conuts". - ("donuts", "coffee"): The company name created is "conuts doffee". - ("donuts", "time"): The company name created is "tonuts dime". - ("donuts", "toffee"): The company name created is "tonuts doffee". - ("time",

”donuts”): The company name created is ”dime tonuts”. - (”toffee”, ”donuts”): The company name created is ”doffee tonuts”. Therefore, there are a total of 6 distinct company names.

The following are some examples of invalid selections: - (”coffee”, ”time”): The name ”toffee” formed after swapping already exists in the original array. - (”time”, ”toffee”): Both names are still the same after swapping and exist in the original array. - (”coffee”, ”toffee”): Both names formed after swapping already exist in the original array.

Example 2: Input: ideas = [”lack”,”back”] Output: 0 Explanation: There are no valid selections. Therefore, 0 is returned.

Constraints:

2 <= ideas.length <= 5 * 104 1 <= ideas[i].length <= 10 ideas[i] consists of lowercase English letters. All the strings in ideas are unique.

==
Problem Number: 2077 URL: https://leetcode.com/problems/selling-pieces-of-wood Title: 2312. Selling Pieces of Wood Problem Description: You are given two integers m and n that represent the height and width of a rectangular piece of wood. You are also given a 2D integer array prices, where prices[i] = [hi, wi, pricei] indicates you can sell a rectangular piece of wood of height hi and width wi for pricei dollars. To cut a piece of wood, you must make a vertical or horizontal cut across the entire height or width of the piece to split it into two smaller pieces. After cutting a piece of wood into some number of smaller pieces, you can sell pieces according to prices. You may sell multiple pieces of the same shape, and you do not have to sell all the shapes. The grain of the wood makes a difference, so you cannot rotate a piece to swap its height and width. Return the maximum money you can earn after cutting an m x n piece of wood. Note that you can cut the piece of wood as many times as you want. Example 1:

Input: m = 3, n = 5, prices = [[1,4,2],[2,2,7],[2,1,3]] Output: 19 Explanation: The diagram above shows a possible scenario. It consists of: - 2 pieces of wood shaped 2 x 2, selling for a price of 2 * 7 = 14. - 1 piece of wood shaped 2 x 1, selling for a price of 1 * 3 = 3. - 1 piece of wood shaped 1 x 4, selling for a price of 1 * 2 = 2. This obtains a total of 14 + 3 + 2 = 19 money earned. It can be shown that 19 is the maximum amount of money that can be earned.

Example 2:

Input: m = 4, n = 6, prices = [[3,2,10],[1,4,2],[4,1,3]] Output: 32 Explanation: The diagram above shows a possible scenario. It consists of: - 3 pieces of wood shaped 3 x 2, selling for a price of 3 * 10 = 30. - 1 piece of wood shaped 1 x 4, selling for a price of 1 * 2 = 2. This obtains a total of 30 + 2 = 32 money earned. It can be shown that 32 is the maximum amount of money that can be earned. Notice that we cannot rotate the 1 x 4 piece of wood to obtain a 4 x 1 piece of wood. Constraints:

1 <= m, n <= 200 1 <= prices.length <= 2 * 104 prices[i].length == 3 1 <= hi <= m 1 <= wi <= n 1 <= pricei <= 106 All the shapes of wood (hi, wi) are pairwise distinct.

==
Problem Number: 2078 URL: https://leetcode.com/problems/number-of-distinct-roll-sequences Title: 2318. Number of Distinct Roll Sequences Problem Description: You are given an integer n. You roll a fair 6-sided dice n times. Determine the total number of distinct sequences of rolls possible such that the following conditions are satisfied:

The greatest common divisor of any adjacent values in the sequence is equal to 1. There is at least a gap of 2 rolls between equal valued rolls. More formally, if the value of the ith roll is equal to the value of the jth roll, then abs(i - j) > 2.

Return the total number of distinct sequences possible. Since the answer may be very large, return it modulo 109 + 7. Two sequences are considered distinct if at least one element is different. Example 1: Input: n = 4 Output: 184 Explanation: Some of the possible sequences are (1, 2, 3, 4), (6, 1, 2, 3), (1, 2, 3, 1), etc. Some invalid sequences are (1, 2, 1, 3), (1, 2, 3, 6). (1, 2, 1, 3) is invalid since the first and third roll have an equal value and abs(1 - 3) = 2 (i and j are 1-indexed). (1, 2, 3, 6) is invalid since the greatest common divisor of 3 and 6 = 3. There are a total of 184 distinct sequences possible, so we return 184. Example 2: Input: n = 2 Output: 22 Explanation: Some of the possible sequences are (1, 2), (2, 1), (3, 2). Some invalid sequences are (3, 6), (2, 4) since the greatest common divisor is not equal to 1. There are a total of 22 distinct sequences possible, so we return 22.

Constraints:

1 <= n <= 104

==
Problem Number: 2079 URL: https://leetcode.com/problems/maximum-score-of-spliced-array Title: 2321. Maximum Score Of Spliced Array Problem Description: You are given two 0-indexed integer arrays nums1 and nums2, both of length n. You can choose two integers left and right where 0 <= left <= right < n and swap the subarray nums1[left...right] with the subarray nums2[left...right].

For example, if nums1 = [1,2,3,4,5] and nums2 = [11,12,13,14,15] and you choose left = 1 and right = 2, nums1 becomes [1,12,13,4,5] and nums2 becomes [11,2,3,14,15].

You may choose to apply the mentioned operation once or not do anything. The score of the arrays is the maximum of sum(nums1) and sum(nums2), where sum(arr) is the sum of all the elements in the array arr. Return the maximum possible score. A subarray is a contiguous sequence of elements within an array. arr[left...right] denotes the subarray that contains the elements of nums between indices left and right (inclusive). Example 1: Input: nums1

= [60,60,60], nums2 = [10,90,10] Output: 210 Explanation: Choosing left = 1 and right = 1, we have nums1 = [60,90,60] and nums2 = [10,60,10]. The score is max(sum(nums1), sum(nums2)) = max(210, 80) = 210. Example 2: Input: nums1 = [20,40,20,70,30], nums2 = [50,20,50,40,20] Output: 220 Explanation: Choosing left = 3, right = 4, we have nums1 = [20,40,20,40,20] and nums2 = [50,20,50,70,30]. The score is max(sum(nums1), sum(nums2)) = max(140, 220) = 220.

Example 3: Input: nums1 = [7,11,13], nums2 = [1,1,1] Output: 31 Explanation: We choose not to swap any subarray. The score is max(sum(nums1), sum(nums2)) = max(31, 3) = 31.

Constraints:

n == nums1.length == nums2.length 1 <= n <= 105 1 <= nums1[i], nums2[i] <= 104

==
Problem Number: 2080 URL: https://leetcode.com/problems/minimum-score-after-removals-on-a-tree Title: 2322. Minimum Score After Removals on a Tree Problem Description: There is an undirected connected tree with n nodes labeled from 0 to n - 1 and n - 1 edges. You are given a 0-indexed integer array nums of length n where nums[i] represents the value of the ith node. You are also given a 2D integer array edges of length n - 1 where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. Remove two distinct edges of the tree to form three connected components. For a pair of removed edges, the following steps are defined:

Get the XOR of all the values of the nodes for each of the three components respectively. The difference between the largest XOR value and the smallest XOR value is the score of the pair.

For example, say the three components have the node values: [4,5,7], [1,9], and [3,3,3]. The three XOR values are 4 ^ 5 ^ 7 = 6, 1 ^ 9 = 8, and 3 ^ 3 ^ 3 = 3. The largest XOR value is 8 and the smallest XOR value is 3. The score is then 8 - 3 = 5.

Return the minimum score of any possible pair of edge removals on the given tree. Example 1:

Input: nums = [1,5,5,4,11], edges = [[0,1],[1,2],[1,3],[3,4]] Output: 9 Explanation: The diagram above shows a way to make a pair of removals. - The 1st component has nodes [1,3,4] with values [5,4,11]. Its XOR value is 5 ^ 4 ^ 11 = 10. - The 2nd component has node [0] with value [1]. Its XOR value is 1 = 1. - The 3rd component has node [2] with value [5]. Its XOR value is 5 = 5. The score is the difference between the largest and smallest XOR value which is 10 - 1 = 9. It can be shown that no other pair of removals will obtain a smaller score than 9.

Example 2:

Input: nums = [5,5,2,4,4,2], edges = [[0,1],[1,2],[5,2],[4,3],[1,3]] Output: 0 Explanation: The diagram above shows a way to make a pair of removals. - The 1st component has nodes [3,4] with values [4,4]. Its XOR value is 4 ^ 4 = 0. - The 2nd component has nodes [1,0] with values [5,5]. Its XOR value is 5 ^ 5 = 0. - The 3rd component has nodes [2,5] with values [2,2]. Its XOR value is 2 ^ 2 = 0. The score is the difference between the largest and smallest XOR value which is 0 - 0 = 0. We cannot obtain a smaller score than 0.

Constraints:

n == nums.length 3 <= n <= 1000 1 <= nums[i] <= 108 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi edges represents a valid tree.

==
Problem Number: 2081 URL: https://leetcode.com/problems/number-of-increasing-paths-in-a-grid Title: 2328. Number of Increasing Paths in a Grid Problem Description: You are given an m x n integer matrix grid, where you can move from a cell to any adjacent cell in all 4 directions. Return the number of strictly increasing paths in the grid such that you can start from any cell and end at any cell. Since the answer may be very large, return it modulo 109 + 7. Two paths are considered different if they do not have exactly the same sequence of visited cells.   Example 1:

Input: grid = [[1,1],[3,4]] Output: 8 Explanation: The strictly increasing paths are: - Paths with length 1: [1], [1], [3], [4]. - Paths with length 2: [1 -> 3], [1 -> 4], [3 -> 4]. - Paths with length 3: [1 -> 3 -> 4]. The total number of paths is 4 + 3 + 1 = 8.

Example 2: Input: grid = [[1],[2]] Output: 3 Explanation: The strictly increasing paths are: - Paths with length 1: [1], [2]. - Paths with length 2: [1 -> 2]. The total number of paths is 2 + 1 = 3.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 1000 1 <= m * n <= 105 1 <= grid[i][j] <= 105

==
Problem Number: 2082 URL: https://leetcode.com/problems/subarray-with-elements-greater-than-varying-threshold Title: 2334. Subarray With Elements Greater Than Varying Threshold Problem Description: You are given an integer array nums and an integer threshold. Find any subarray of nums of length k such that every element in the subarray is greater than threshold / k. Return the size of any such subarray. If there is no such subarray, return -1. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,3,4,3,1], threshold = 6 Output: 3 Explanation: The subarray [3,4,3] has a size of 3, and every element is greater than 6 / 3 = 2. Note that this is the only valid subarray.

Example 2: Input: nums = [6,5,6,5,8], threshold = 7 Output: 1 Explanation:

The subarray [8] has a size of 1, and 8 > 7 / 1 = 7. So 1 is returned. Note that the subarray [6,5] has a size of 2, and every element is greater than 7 / 2 = 3.5. Similarly, the subarrays [6,5,6], [6,5,6,5], [6,5,6,5,8] also satisfy the given conditions. Therefore, 2, 3, 4, or 5 may also be returned. Constraints:

1 <= nums.length <= 105 1 <= nums[i], threshold <= 109

==
Problem Number: 2083 URL: https://leetcode.com/problems/count-the-number-of-ideal-arrays Title: 2338. Count the Number of Ideal Arrays Problem Description: You are given two integers n and maxValue, which are used to describe an ideal array. A 0-indexed integer array arr of length n is considered ideal if the following conditions hold:

Every arr[i] is a value from 1 to maxValue, for 0 <= i < n. Every arr[i] is divisible by arr[i - 1], for 0 < i < n.

Return the number of distinct ideal arrays of length n. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: n = 2, maxValue = 5 Output: 10 Explanation: The following are the possible ideal arrays: - Arrays starting with the value 1 (5 arrays): [1,1], [1,2], [1,3], [1,4], [1,5] - Arrays starting with the value 2 (2 arrays): [2,2], [2,4] - Arrays starting with the value 3 (1 array): [3,3] - Arrays starting with the value 4 (1 array): [4,4] - Arrays starting with the value 5 (1 array): [5,5] There are a total of 5 + 2 + 1 + 1 + 1 = 10 distinct ideal arrays.

Example 2: Input: n = 5, maxValue = 3 Output: 11 Explanation: The following are the possible ideal arrays: - Arrays starting with the value 1 (9 arrays): - With no other distinct values (1 array): [1,1,1,1,1] - With 2nd distinct value 2 (4 arrays): [1,1,1,1,2], [1,1,1,2,2], [1,1,2,2,2], [1,2,2,2,2] - With 2nd distinct value 3 (4 arrays): [1,1,1,1,3], [1,1,1,3,3], [1,1,3,3,3], [1,3,3,3,3] - Arrays starting with the value 2 (1 array): [2,2,2,2,2] - Arrays starting with the value 3 (1 array): [3,3,3,3,3] There are a total of 9 + 1 + 1 = 11 distinct ideal arrays.

Constraints:

2 <= n <= 104 1 <= maxValue <= 104

==
Problem Number: 2084 URL: https://leetcode.com/problems/minimum-deletions-to-make-array-divisible Title: 2344. Minimum Deletions to Make Array Divisible Problem Description: You are given two positive integer arrays nums and numsDivide. You can delete any number of elements from nums. Return the minimum number of deletions such that the smallest element in nums divides all the elements of numsDivide. If this is not possible, return -1. Note that an integer x divides y if y % x == 0. Example 1: Input: nums = [2,3,2,4,3], numsDivide = [9,6,9,3,15] Output: 2 Explanation: The smallest element in [2,3,2,4,3] is 2, which does not divide all the elements of numsDivide. We use 2 deletions to delete the elements in nums that are equal to 2 which makes nums = [3,4,3]. The smallest element in [3,4,3] is 3, which divides all

the elements of numsDivide. It can be shown that 2 is the minimum number of deletions needed.

Example 2: Input: nums = [4,3,6], numsDivide = [8,2,6,10] Output: -1 Explanation: We want the smallest element in nums to divide all the elements of numsDivide. There is no way to delete elements from nums to allow this. Constraints:

1 <= nums.length, numsDivide.length <= 105 1 <= nums[i], numsDivide[i] <= 109

==

Problem Number: 2085 URL: https://leetcode.com/problems/shortest-impossible-sequence-of-rolls Title: 2350. Shortest Impossible Sequence of Rolls Problem Description: You are given an integer array rolls of length n and an integer k. You roll a k sided dice numbered from 1 to k, n times, where the result of the ith roll is rolls[i]. Return the length of the shortest sequence of rolls that cannot be taken from rolls. A sequence of rolls of length len is the result of rolling a k sided dice len times. Note that the sequence taken does not have to be consecutive as long as it is in order. Example 1: Input: rolls = [4,2,1,2,3,3,2,4,1], k = 4 Output: 3 Explanation: Every sequence of rolls of length 1, [1], [2], [3], [4], can be taken from rolls. Every sequence of rolls of length 2, [1, 1], [1, 2], ..., [4, 4], can be taken from rolls. The sequence [1, 4, 2] cannot be taken from rolls, so we return 3. Note that there are other sequences that cannot be taken from rolls. Example 2: Input: rolls = [1,1,2,2], k = 2 Output: 2 Explanation: Every sequence of rolls of length 1, [1], [2], can be taken from rolls. The sequence [2, 1] cannot be taken from rolls, so we return 2. Note that there are other sequences that cannot be taken from rolls but [2, 1] is the shortest.

Example 3: Input: rolls = [1,1,3,2,2,2,3,3], k = 4 Output: 1 Explanation: The sequence [4] cannot be taken from rolls, so we return 1. Note that there are other sequences that cannot be taken from rolls but [4] is the shortest.

Constraints:

n == rolls.length 1 <= n <= 105 1 <= rolls[i] <= k <= 105

==

Problem Number: 2086 URL: https://leetcode.com/problems/number-of-excellent-pairs Title: 2354. Number of Excellent Pairs Problem Description: You are given a 0-indexed positive integer array nums and a positive integer k. A pair of numbers (num1, num2) is called excellent if the following conditions are satisfied:

Both the numbers num1 and num2 exist in the array nums. The sum of the number of set bits in num1 OR num2 and num1 AND num2 is greater than or equal to k, where OR is the bitwise OR operation and AND is the bitwise AND operation.

Return the number of distinct excellent pairs. Two pairs (a, b) and (c, d) are considered distinct if either a != c or b != d. For example, (1, 2) and (2, 1) are distinct. Note that a pair (num1, num2) such that num1 == num2 can also be excellent if you have at least one occurrence of num1 in the array. Example 1: Input: nums = [1,2,3,1], k = 3 Output: 5 Explanation: The excellent pairs are the following: - (3, 3). (3 AND 3) and (3 OR 3) are both equal to (11) in binary. The total number of set bits is 2 + 2 = 4, which is greater than or equal to k = 3. - (2, 3) and (3, 2). (2 AND 3) is equal to (10) in binary, and (2 OR 3) is equal to (11) in binary. The total number of set bits is 1 + 2 = 3. - (1, 3) and (3, 1). (1 AND 3) is equal to (01) in binary, and (1 OR 3) is equal to (11) in binary. The total number of set bits is 1 + 2 = 3. So the number of excellent pairs is 5. Example 2: Input: nums = [5,1,1], k = 10 Output: 0 Explanation: There are no excellent pairs for this array.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109 1 <= k <= 60

==

Problem Number: 2087 URL: https://leetcode.com/problems/longest-cycle-in-a-graph Title: 2360. Longest Cycle in a Graph Problem Description: You are given a directed graph of n nodes numbered from 0 to n - 1, where each node has at most one outgoing edge. The graph is represented with a given 0-indexed array edges of size n, indicating that there is a directed edge from node i to node edges[i]. If there is no outgoing edge from node i, then edges[i] == -1. Return the length of the longest cycle in the graph. If no cycle exists, return -1. A cycle is a path that starts and ends at the same node. Example 1:

Input: edges = [3,3,4,2,3] Output: 3 Explanation: The longest cycle in the graph is the cycle: 2 -> 4 -> 3 -> 2. The length of this cycle is 3, so 3 is returned.

Example 2:

Input: edges = [2,-1,3,1] Output: -1 Explanation: There are no cycles in this graph.

Constraints:

n == edges.length 2 <= n <= 105 -1 <= edges[i] < n edges[i] != i

==

Problem Number: 2088 URL: https://leetcode.com/problems/minimum-replacements-to-sort-the-array Title: 2366. Minimum Replacements to Sort the Array Problem Description: You are given a 0-indexed integer array nums. In one operation you can replace any element of the array with any two elements that sum to it.

For example, consider nums = [5,6,7]. In one operation, we can replace nums[1] with 2 and 4 and convert nums to [5,2,4,7].

Return the minimum number of operations to make an array that is sorted in non-decreasing order. Example 1: Input: nums = [3,9,3] Output: 2 Explanation: Here are the steps to sort the array in non-decreasing order: - From [3,9,3], replace the 9 with 3 and 6 so the array becomes [3,3,6,3] - From [3,3,6,3], replace the 6 with 3 and 3 so the array becomes [3,3,3,3,3] There are 2 steps to sort the array in non-decreasing order. Therefore, we return 2.

Example 2: Input: nums = [1,2,3,4,5] Output: 0 Explanation: The array is already in non-decreasing order. Therefore, we return 0.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==

Problem Number: 2089 URL: https://leetcode.com/problems/count-special-integers Title: 2376. Count Special Integers Problem Description: We call a positive integer special if all of its digits are distinct. Given a positive integer n, return the number of special integers that belong to the interval [1, n]. Example 1: Input: n = 20 Output: 19 Explanation: All the integers from 1 to 20, except 11, are special. Thus, there are 19 special integers.

Example 2: Input: n = 5 Output: 5 Explanation: All the integers from 1 to 5 are special.

Example 3: Input: n = 135 Output: 110 Explanation: There are 110 integers from 1 to 135 that are special. Some of the integers that are not special are: 22, 114, and 131. Constraints:

1 <= n <= 2 * 109

==

Problem Number: 2090 URL: https://leetcode.com/problems/maximum-segment-sum-after-removals Title: 2382. Maximum Segment Sum After Removals Problem Description: You are given two 0-indexed integer arrays nums and removeQueries, both of length n. For the ith query, the element in nums at the index removeQueries[i] is removed, splitting nums into different segments. A segment is a contiguous sequence of positive integers in nums. A segment sum is the sum of every element in a segment. Return an integer array answer, of length n, where answer[i] is the maximum segment sum after applying the ith removal. Note: The same index will not be removed more than once. Example 1: Input: nums = [1,2,5,6,1], removeQueries = [0,3,2,4,1] Output: [14,7,2,2,0] Explanation: Using 0 to indicate a removed element, the answer is as follows: Query 1: Remove the 0th element, nums becomes [0,2,5,6,1] and the maximum segment sum is 14 for segment [2,5,6,1]. Query 2: Remove the 3rd element, nums becomes [0,2,5,0,1] and the maximum segment sum is 7 for segment [2,5]. Query 3: Remove the 2nd element, nums becomes [0,2,0,0,1] and the maximum segment sum is 2 for segment [2]. Query 4: Remove the 4th element, nums becomes [0,2,0,0,0] and the maximum segment sum is 2 for segment [2]. Query 5: Remove the 1st element, nums becomes [0,0,0,0,0] and the maximum segment sum is 0, since there are no segments. Finally, we return [14,7,2,2,0]. Example 2: Input: nums = [3,2,11,1], removeQueries = [3,2,1,0] Output: [16,5,3,0] Explanation: Using 0 to indicate a removed element, the answer is as follows: Query 1: Remove the 3rd element, nums becomes [3,2,11,0] and the maximum segment sum is 16 for segment [3,2,11]. Query 2: Remove the 2nd element, nums becomes [3,2,0,0] and the maximum segment sum is 5 for segment [3,2]. Query 3: Remove the 1st element, nums becomes [3,0,0,0] and the maximum segment sum is 3 for segment [3]. Query 4: Remove the 0th element, nums becomes [0,0,0,0] and the maximum segment sum is 0, since there are no segments. Finally, we return [16,5,3,0].

Constraints:

n == nums.length == removeQueries.length 1 <= n <= 105 1 <= nums[i] <= 109 0 <= removeQueries[i] < n All the values of removeQueries are unique.

==
Problem Number: 2091 URL: https://leetcode.com/problems/find-the-k-sum-of-an-array Title: 2386. Find the K-Sum of an Array Problem Description: You are given an integer array nums and a positive integer k. You can choose any subsequence of the array and sum all of its elements together. We define the K-Sum of the array as the kth largest subsequence sum that can be obtained (not necessarily distinct). Return the K-Sum of the array. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements. Note that the empty subsequence is considered to have a sum of 0. Example 1: Input: nums = [2,4,-2], k = 5 Output: 2 Explanation: All the possible subsequence sums that we can obtain are the following sorted in decreasing order: - 6, 4, 4, 2, 2, 0, 0, -2. The 5-Sum of the array is 2.

Example 2: Input: nums = [1,-2,3,4,-10,12], k = 16 Output: 10 Explanation: The 16-Sum of the array is 10.

Constraints:

n == nums.length 1 <= n <= 105 -109 <= nums[i] <= 109 1 <= k <= min(2000, 2n)

==
Problem Number: 2092 URL: https://leetcode.com/problems/build-a-matrix-with-conditions Title: 2392. Build a Matrix With Conditions Problem Description: You are given a positive integer k. You are also given:

a 2D integer array rowConditions of size n where rowConditions[i] = [abovei, belowi], and a 2D integer array colConditions of size m where colConditions[i] = [lefti, righti].

The two arrays contain integers from 1 to k. You have to build a k x k matrix that contains each of the numbers from 1 to k exactly once. The remaining cells should have the value 0. The matrix should also satisfy the following conditions:

The number abovei should appear in a row that is strictly above the row at which the number belowi appears for all i from 0 to n - 1. The number lefti should appear in a column that is strictly left of the column at which the number righti appears for all i from 0 to m - 1.

Return any matrix that satisfies the conditions. If no answer exists, return an empty matrix. Example 1:

Input: k = 3, rowConditions = [[1,2],[3,2]], colConditions = [[2,1],[3,2]] Output: [[3,0,0],[0,0,1],[0,2,0]] Explanation: The diagram above shows a valid example of a matrix that satisfies all the conditions. The row conditions are the following: - Number 1 is in row 1, and number 2 is in row 2, so 1 is above 2 in the matrix. - Number 3 is in row 0, and number 2 is in row 2, so 3 is above 2 in the matrix. The column conditions are the following: - Number 2 is in column 1, and number 1 is in column 2, so 2 is left of 1 in the matrix. - Number 3 is in column 0, and number 2 is in column 1, so 3 is left of 2 in the matrix. Note that there may be multiple correct answers.

Example 2: Input: k = 3, rowConditions = [[1,2],[2,3],[3,1],[2,3]], colConditions = [[2,1]] Output: [] Explanation: From the first two conditions, 3 has to be below 1 but the third conditions needs 3 to be above 1 to be satisfied. No matrix can satisfy all the conditions, so we return the empty matrix.

Constraints:

2 <= k <= 400 1 <= rowConditions.length, colConditions.length <= 104 rowConditions[i].length == colConditions[i].length == 2 1 <= abovei, belowi, lefti, righti <= k abovei != belowi lefti != righti

==
Problem Number: 2093 URL: https://leetcode.com/problems/maximum-number-of-robots-within-budget Title: 2398. Maximum Number of Robots Within Budget Problem Description: You have n robots. You are given two 0-indexed integer arrays, chargeTimes and runningCosts, both of length n. The ith robot costs chargeTimes[i] units to charge and costs runningCosts[i] units to run. You are also given an integer budget. The total cost of running k chosen robots is equal to max(chargeTimes) + k * sum(runningCosts), where max(chargeTimes) is the largest charge cost among the k robots and sum(runningCosts) is the sum of running costs among the k robots. Return the maximum number of consecutive robots you can run such that the total cost does not exceed budget. Example 1: Input: chargeTimes = [3,6,1,3,4], runningCosts = [2,1,3,4,5], budget = 25 Output: 3 Explanation: It is possible to run all individual and consecutive pairs of robots within budget. To obtain answer 3, consider the first 3 robots. The total cost will be max(3,6,1) + 3 * sum(2,1,3) = 6 + 3 * 6 = 24 which is less than 25. It can be shown that it is not possible to run more than 3 consecutive robots within budget, so we return 3.

Example 2: Input: chargeTimes = [11,12,19], runningCosts = [10,8,7], budget

= 19 Output: 0 Explanation: No robot can be run that does not exceed the budget, so we return 0.

Constraints:

chargeTimes.length == runningCosts.length == n 1 <= n <= 5 * 104 1 <= chargeTimes[i], runningCosts[i] <= 105 1 <= budget <= 1015

==

Problem Number: 2094 URL: https://leetcode.com/problems/meeting-rooms-iii Title: 2402. Meeting Rooms III Problem Description: You are given an integer n. There are n rooms numbered from 0 to n - 1. You are given a 2D integer array meetings where meetings[i] = [starti, endi] means that a meeting will be held during the half-closed time interval [starti, endi). All the values of starti are unique. Meetings are allocated to rooms in the following manner:

Each meeting will take place in the unused room with the lowest number. If there are no available rooms, the meeting will be delayed until a room becomes free. The delayed meeting should have the same duration as the original meeting. When a room becomes unused, meetings that have an earlier original start time should be given the room.

Return the number of the room that held the most meetings. If there are multiple rooms, return the room with the lowest number. A half-closed interval [a, b) is the interval between a and b including a and not including b.   Example 1: Input: n = 2, meetings = [[0,10],[1,5],[2,7],[3,4]] Output: 0 Explanation: - At time 0, both rooms are not being used. The first meeting starts in room 0. - At time 1, only room 1 is not being used. The second meeting starts in room 1. - At time 2, both rooms are being used. The third meeting is delayed. - At time 3, both rooms are being used. The fourth meeting is delayed. - At time 5, the meeting in room 1 finishes. The third meeting starts in room 1 for the time period [5,10). - At time 10, the meetings in both rooms finish. The fourth meeting starts in room 0 for the time period [10,11). Both rooms 0 and 1 held 2 meetings, so we return 0.

Example 2: Input: n = 3, meetings = [[1,20],[2,10],[3,5],[4,9],[6,8]] Output: 1 Explanation: - At time 1, all three rooms are not being used. The first meeting starts in room 0. - At time 2, rooms 1 and 2 are not being used. The second meeting starts in room 1. - At time 3, only room 2 is not being used. The third meeting starts in room 2. - At time 4, all three rooms are being used. The fourth meeting is delayed. - At time 5, the meeting in room 2 finishes. The fourth meeting starts in room 2 for the time period [5,10). - At time 6, all three rooms are being used. The fifth meeting is delayed. - At time 10, the meetings in rooms 1 and 2 finish. The fifth meeting starts in room 1 for the time period [10,12). Room 0 held 1 meeting while rooms 1 and 2 each held 2 meetings, so we return 1.

Constraints:

1 <= n <= 100 1 <= meetings.length <= 105 meetings[i].length == 2 0 <= starti < endi <= 5 * 105 All the values of starti are unique.

==

Problem Number: 2095 URL: https://leetcode.com/problems/longest-increasing-subsequence-ii Title: 2407. Longest Increasing Subsequence II Problem Description: You are given an integer array nums and an integer k. Find the longest subsequence of nums that meets the following requirements:

The subsequence is strictly increasing and The difference between adjacent elements in the subsequence is at most k.

Return the length of the longest subsequence that meets the requirements. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements. Example 1: Input: nums = [4,2,1,4,3,4,5,8,15], k = 3 Output: 5 Explanation: The longest subsequence that meets the requirements is [1,3,4,5,8]. The subsequence has a length of 5, so we return 5. Note that the subsequence [1,3,4,5,8,15] does not meet the requirements because 15 - 8 = 7 is larger than 3.

Example 2: Input: nums = [7,4,5,1,8,12,4,7], k = 5 Output: 4 Explanation: The longest subsequence that meets the requirements is [4,5,8,12]. The subsequence has a length of 4, so we return 4.

Example 3: Input: nums = [1,5], k = 1 Output: 1 Explanation: The longest subsequence that meets the requirements is [1]. The subsequence has a length of 1, so we return 1.

Constraints:

1 <= nums.length <= 105 1 <= nums[i], k <= 105

==

Problem Number: 2096 URL: https://leetcode.com/problems/minimum-money-required-before-transactions Title: 2412. Minimum Money Required Before Transactions Problem Description: You are given a 0-indexed 2D integer array transactions, where transactions[i] = [costi, cashbacki]. The array describes transactions, where each transaction must be completed exactly once in some order. At any given moment, you have a certain amount of money. In order to complete transaction i, money >= costi must hold true. After performing a transaction, money becomes money - costi + cashbacki. Return the minimum amount of money required before any transaction so that all of the transactions can be completed regardless of the order of the transactions. Example 1: Input: transactions = [[2,1],[5,0],[4,2]] Output: 10 Explanation: Starting with money = 10, the transactions can be performed in any order. It can be shown that starting with money < 10 will fail to complete all transactions in some order.

Example 2: Input: transactions = [[3,0],[0,3]] Output: 3 Explanation: - If transactions are in the order [[3,0],[0,3]], the minimum money required to complete the transactions is 3. - If transactions are in the order [[0,3],[3,0]], the minimum money required to complete the transactions is 0. Thus, starting with money = 3, the transactions can be performed in any order.

Constraints:

1 <= transactions.length <= 105 transactions[i].length == 2 0 <= costi, cashbacki <= 109

==
Problem Number: 2097 URL: https://leetcode.com/problems/sum-of-prefix-scores-of-strings Title: 2416. Sum of Prefix Scores of Strings Problem Description: You are given an array words of size n consisting of non-empty strings. We define the score of a string word as the number of strings words[i] such that word is a prefix of words[i].

For example, if words = ["a", "ab", "abc", "cab"], then the score of "ab" is 2, since "ab" is a prefix of both "ab" and "abc".

Return an array answer of size n where answer[i] is the sum of scores of every non-empty prefix of words[i]. Note that a string is considered as a prefix of itself. Example 1: Input: words = ["abc","ab","bc","b"] Output: [5,4,3,2] Explanation: The answer for each string is the following: - "abc" has 3 prefixes: "a", "ab", and "abc". - There are 2 strings with the prefix "a", 2 strings with the prefix "ab", and 1 string with the prefix "abc". The total is answer[0] = 2 + 2 + 1 = 5. - "ab" has 2 prefixes: "a" and "ab". - There are 2 strings with the prefix "a", and 2 strings with the prefix "ab". The total is answer[1] = 2 + 2 = 4. - "bc" has 2 prefixes: "b" and "bc". - There are 2 strings with the prefix "b", and 1 string with the prefix "bc". The total is answer[2] = 2 + 1 = 3. - "b" has 1 prefix: "b". - There are 2 strings with the prefix "b". The total is answer[3] = 2.

Example 2: Input: words = ["abcd"] Output: [4] Explanation: "abcd" has 4 prefixes: "a", "ab", "abc", and "abcd". Each prefix has a score of one, so the total is answer[0] = 1 + 1 + 1 + 1 = 4.

Constraints:

1 <= words.length <= 1000 1 <= words[i].length <= 1000 words[i] consists of lowercase English letters.

==
Problem Number: 2098 URL: https://leetcode.com/problems/number-of-good-paths Title: 2421. Number of Good Paths Problem Description: There is a tree (i.e. a connected, undirected graph with no cycles) consisting of n nodes numbered from 0 to n - 1 and exactly n - 1 edges. You are given a 0-indexed integer array vals of length n where vals[i] denotes the value of the ith node. You are also given a 2D integer array edges where edges[i] = [ai, bi] denotes that there exists an undirected edge connecting nodes ai and bi. A good path is a simple path that satisfies the following conditions:

The starting node and the ending node have the same value. All nodes between the starting node and the ending node have values less than or equal to the starting node (i.e. the starting node's value should be the maximum value along the path).

Return the number of distinct good paths. Note that a path and its reverse are counted as the same path. For example, 0 -> 1 is considered to be the same as 1 -> 0. A single node is also considered as a valid path.   Example 1:

Input: vals = [1,3,2,1,3], edges = [[0,1],[0,2],[2,3],[2,4]] Output: 6 Explanation: There are 5 good paths consisting of a single node. There is 1 additional good path: 1 -> 0 -> 2 -> 4. (The reverse path 4 -> 2 -> 0 -> 1 is treated as the same as 1 -> 0 -> 2 -> 4.) Note that 0 -> 2 -> 3 is not a good path because vals[2] > vals[0].

Example 2:

Input: vals = [1,1,2,2,3], edges = [[0,1],[1,2],[2,3],[2,4]] Output: 7 Explanation: There are 5 good paths consisting of a single node. There are 2 additional good paths: 0 -> 1 and 2 -> 3.

Example 3:

Input: vals = [1], edges = [] Output: 1 Explanation: The tree consists of only one node, so there is one good path.

Constraints:

n == vals.length 1 <= n <= 3 * 104 0 <= vals[i] <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi edges represents a valid tree.

==

Problem Number: 2099 URL: https://leetcode.com/problems/number-of-pairs-satisfying-inequality Title: 2426. Number of Pairs Satisfying Inequality Problem Description: You are given two 0-indexed integer arrays nums1 and nums2, each of size n, and an integer diff. Find the number of pairs (i, j) such that:

0 <= i < j <= n - 1 and nums1[i] - nums1[j] <= nums2[i] - nums2[j] + diff.

Return the number of pairs that satisfy the conditions.   Example 1: Input: nums1 = [3,2,5], nums2 = [2,2,1], diff = 1 Output: 3 Explanation: There are 3 pairs that satisfy the conditions: 1. i = 0, j = 1: 3 - 2 <= 2 - 2 + 1. Since i < j and 1 <= 1, this pair satisfies the conditions. 2. i = 0, j = 2: 3 - 5 <= 2 - 1 + 1. Since i < j and -2 <= 2, this pair satisfies the conditions. 3. i = 1, j = 2: 2 - 5 <= 2 - 1 + 1. Since i < j and -3 <= 2, this pair satisfies the conditions. Therefore, we return 3.

Example 2: Input: nums1 = [3,-1], nums2 = [-2,2], diff = -1 Output: 0 Explanation: Since there does not exist any pair that satisfies the conditions, we return 0.

Constraints:

n == nums1.length == nums2.length 2 <= n <= 105 -104 <= nums1[i], nums2[i] <= 104 -104 <= diff <= 104

==
Problem Number: 2100 URL: https://leetcode.com/problems/maximum-deletions-on-a-string Title: 2430. Maximum Deletions on a String Problem Description: You are given a string s consisting of only lowercase English letters. In one operation, you can:

Delete the entire string s, or Delete the first i letters of s if the first i letters of s are equal to the following i letters in s, for any i in the range 1 <= i <= s.length / 2.

For example, if s = "ababc", then in one operation, you could delete the first two letters of s to get "abc", since the first two letters of s and the following two letters of s are both equal to "ab". Return the maximum number of operations needed to delete all of s. Example 1: Input: s = "abcabcdabc" Output: 2 Explanation: - Delete the first 3 letters ("abc") since the next 3 letters are equal. Now, s = "abcdabc". - Delete all the letters. We used 2 operations so return 2. It can be proven that 2 is the maximum number of operations needed. Note that in the second operation we cannot delete "abc" again because the next occurrence of "abc" does not happen in the next 3 letters.

Example 2: Input: s = "aaabaab" Output: 4 Explanation: - Delete the first letter ("a") since the next letter is equal. Now, s = "aabaab". - Delete the first 3 letters ("aab") since the next 3 letters are equal. Now, s = "aab". - Delete the first letter ("a") since the next letter is equal. Now, s = "ab". - Delete all the letters. We used 4 operations so return 4. It can be proven that 4 is the maximum number of operations needed.

Example 3: Input: s = "aaaaa" Output: 5 Explanation: In each operation, we can delete the first letter of s.

Constraints:

1 <= s.length <= 4000 s consists only of lowercase English letters.

==
Problem Number: 2101 URL: https://leetcode.com/problems/paths-in-matrix-whose-sum-is-divisible-by-k Title: 2435. Paths in Matrix Whose Sum Is Divisible by K Problem Description: You are given a 0-indexed m x n integer matrix grid and an integer k. You are currently at position (0, 0) and you want to reach position (m - 1, n - 1) moving only down or right. Return the number of paths where the sum of the elements on the path is divisible by k. Since the answer may be very large, return it modulo 109 + 7. Example 1:

Input: grid = [[5,2,4],[3,0,5],[0,7,2]], k = 3 Output: 2 Explanation: There are two paths where the sum of the elements on the path is divisible by k. The first path highlighted in red has a sum of 5 + 2 + 4 + 5 + 2 = 18 which is divisible by 3. The second path highlighted in blue has a sum of 5 + 3 + 0 + 5 + 2 = 15 which is divisible by 3.

Example 2:

Input: grid = [[0,0]], k = 5 Output: 1 Explanation: The path highlighted in red has a sum of 0 + 0 = 0 which is divisible by 5.

Example 3:

Input: grid = [[7,3,4,9],[2,3,6,2],[2,3,7,0]], k = 1 Output: 10 Explanation: Every integer is divisible by 1 so the sum of the elements on every possible path is divisible by k.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 5 * 104 1 <= m * n <= 5 * 104 0 <= grid[i][j] <= 100 1 <= k <= 50

==
Problem Number: 2102 URL: https://leetcode.com/problems/create-components-with-same-value Title: 2440. Create Components With Same Value Problem Description: There is an undirected tree with n nodes labeled from 0 to n - 1. You are given a 0-indexed integer array nums of length n where nums[i] represents the value of the ith node. You are also given a 2D integer array edges of length n - 1 where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. You are allowed to delete some edges, splitting the tree into multiple connected components. Let the value of a component be the sum of all nums[i] for which node i is in the component. Return the maximum number of edges you can delete, such that every connected component in the tree has the same value. Example 1:

Input: nums = [6,2,2,2,6], edges = [[0,1],[1,2],[1,3],[3,4]] Output: 2 Explanation: The above figure shows how we can delete the edges [0,1] and [3,4]. The created components are nodes [0], [1,2,3] and [4]. The sum of the values in each component equals 6. It can be proven that no better deletion exists, so the answer is 2.

Example 2: Input: nums = [2], edges = [] Output: 0 Explanation: There are no edges to be deleted.

Constraints:

1 <= n <= 2 * 104 nums.length == n 1 <= nums[i] <= 50 edges.length == n - 1 edges[i].length == 2 0 <= edges[i][0], edges[i][1] <= n - 1 edges represents a valid tree.

==
Problem Number: 2103 URL: https://leetcode.com/problems/count-subarrays-with-fixed-bounds Title: 2444. Count Subarrays With Fixed Bounds Problem Description: You are given an integer array nums and two integers minK and

maxK. A fixed-bound subarray of nums is a subarray that satisfies the following conditions:

The minimum value in the subarray is equal to minK. The maximum value in the subarray is equal to maxK.

Return the number of fixed-bound subarrays. A subarray is a contiguous part of an array. Example 1: Input: nums = [1,3,5,2,7,5], minK = 1, maxK = 5 Output: 2 Explanation: The fixed-bound subarrays are [1,3,5] and [1,3,5,2].

Example 2: Input: nums = [1,1,1,1], minK = 1, maxK = 1 Output: 10 Explanation: Every subarray of nums is a fixed-bound subarray. There are 10 possible subarrays.

Constraints:

2 <= nums.length <= 105 1 <= nums[i], minK, maxK <= 106

==
Problem Number: 2104 URL: https://leetcode.com/problems/minimum-cost-to-make-array-equal Title: 2448. Minimum Cost to Make Array Equal Problem Description: You are given two 0-indexed arrays nums and cost consisting each of n positive integers. You can do the following operation any number of times:

Increase or decrease any element of the array nums by 1.

The cost of doing one operation on the ith element is cost[i]. Return the minimum total cost such that all the elements of the array nums become equal. Example 1: Input: nums = [1,3,5,2], cost = [2,3,1,14] Output: 8 Explanation: We can make all the elements equal to 2 in the following way: - Increase the 0th element one time. The cost is 2. - Decrease the 1st element one time. The cost is 3. - Decrease the 2nd element three times. The cost is 1 + 1 + 1 = 3. The total cost is 2 + 3 + 3 = 8. It can be shown that we cannot make the array equal with a smaller cost.

Example 2: Input: nums = [2,2,2,2,2], cost = [4,2,8,1,3] Output: 0 Explanation: All the elements are already equal, so no operations are needed.

Constraints:

n == nums.length == cost.length 1 <= n <= 105 1 <= nums[i], cost[i] <= 106 Test cases are generated in a way that the output doesn't exceed 253-1

==
Problem Number: 2105 URL: https://leetcode.com/problems/minimum-number-of-operations-to-make-arrays-similar Title: 2449. Minimum Number of Operations to Make Arrays Similar Problem Description: You are given two positive integer arrays nums and target, of the same length. In one operation, you can choose any two distinct indices i and j where 0 <= i, j < nums.length and:

set nums[i] = nums[i] + 2 and set nums[j] = nums[j] - 2.

Two arrays are considered to be similar if the frequency of each element is the same. Return the minimum number of operations required to make nums similar to target. The test cases are generated such that nums can always be similar to target.   Example 1: Input: nums = [8,12,6], target = [2,14,10] Output: 2 Explanation: It is possible to make nums similar to target in two operations: - Choose i = 0 and j = 2, nums = [10,12,4]. - Choose i = 1 and j = 2, nums = [10,14,2]. It can be shown that 2 is the minimum number of operations needed.

Example 2: Input: nums = [1,2,5], target = [4,1,3] Output: 1 Explanation: We can make nums similar to target in one operation: - Choose i = 1 and j = 2, nums = [1,4,3].

Example 3: Input: nums = [1,1,1,1,1], target = [1,1,1,1,1] Output: 0 Explanation: The array nums is already similiar to target.

Constraints:

n == nums.length == target.length 1 <= n <= 105 1 <= nums[i], target[i] <= 106 It is possible to make nums similar to target.

==
Problem Number: 2106 URL: https://leetcode.com/problems/next-greater-element-iv Title: 2454. Next Greater Element IV Problem Description: You are given a 0-indexed array of non-negative integers nums. For each integer in nums, you must find its respective second greater integer. The second greater integer of nums[i] is nums[j] such that:

j > i nums[j] > nums[i] There exists exactly one index k such that nums[k] > nums[i] and i < k < j.

If there is no such nums[j], the second greater integer is considered to be -1.

For example, in the array [1, 2, 4, 3], the second greater integer of 1 is 4, 2 is 3, and that of 3 and 4 is -1.

Return an integer array answer, where answer[i] is the second greater integer of nums[i].   Example 1: Input: nums = [2,4,0,9,6] Output: [9,6,6,-1,-1] Explanation: 0th index: 4 is the first integer greater than 2, and 9 is the second integer greater than 2, to the right of 2. 1st index: 9 is the first, and 6 is the second integer greater than 4, to the right of 4. 2nd index: 9 is the first, and 6 is the second integer greater than 0, to the right of 0. 3rd index: There is no integer greater than 9 to its right, so the second greater integer is considered to be -1. 4th index: There is no integer greater than 6 to its right, so the second greater integer is considered to be -1. Thus, we return [9,6,6,-1,-1].

Example 2: Input: nums = [3,3] Output: [-1,-1] Explanation: We return [-1,-1] since neither integer has any integer greater than it.

Constraints:

1 <= nums.length <= 105 0 <= nums[i] <= 109

==
Problem Number: 2107 URL: https://leetcode.com/problems/height-of-binary-tree-after-subtree-removal-queries Title: 2458. Height of Binary Tree After Subtree Removal Queries Problem Description: You are given the root of a binary tree with n nodes. Each node is assigned a unique value from 1 to n. You are also given an array queries of size m. You have to perform m independent queries on the tree where in the ith query you do the following:

Remove the subtree rooted at the node with the value queries[i] from the tree. It is guaranteed that queries[i] will not be equal to the value of the root.

Return an array answer of size m where answer[i] is the height of the tree after performing the ith query. Note:

The queries are independent, so the tree returns to its initial state after each query. The height of a tree is the number of edges in the longest simple path from the root to some node in the tree.

Example 1:

Input: root = [1,3,4,2,null,6,5,null,null,null,null,null,7], queries = [4] Output: [2] Explanation: The diagram above shows the tree after removing the subtree rooted at node with value 4. The height of the tree is 2 (The path 1 -> 3 -> 2).

Example 2:

Input: root = [5,8,9,2,1,3,7,4,6], queries = [3,2,4,8] Output: [3,2,3,2] Explanation: We have the following queries: - Removing the subtree rooted at node with value 3. The height of the tree becomes 3 (The path 5 -> 8 -> 2 -> 4). - Removing the subtree rooted at node with value 2. The height of the tree becomes 2 (The path 5 -> 8 -> 1). - Removing the subtree rooted at node with value 4. The height of the tree becomes 3 (The path 5 -> 8 -> 2 -> 6). - Removing the subtree rooted at node with value 8. The height of the tree becomes 2 (The path 5 -> 9 -> 3).

Constraints:

The number of nodes in the tree is n. 2 <= n <= 105 1 <= Node.val <= n All the values in the tree are unique. m == queries.length 1 <= m <= min(n, 104) 1 <= queries[i] <= n queries[i] != root.val

==
Problem Number: 2108 URL: https://leetcode.com/problems/minimum-total-distance-traveled Title: 2463. Minimum Total Distance Traveled Problem Description: There are some robots and factories on the X-axis. You are given an integer array robot where robot[i] is the position of the ith robot. You are also given a 2D integer array factory where factory[j] = [positionj, limitj] indicates that positionj is the position of the jth factory and that the jth factory can repair at most limitj robots. The positions of each robot are unique. The positions of each factory are also unique. Note that a robot can be in the same position as a factory initially. All the robots are initially broken; they keep

moving in one direction. The direction could be the negative or the positive direction of the X-axis. When a robot reaches a factory that did not reach its limit, the factory repairs the robot, and it stops moving. At any moment, you can set the initial direction of moving for some robot. Your target is to minimize the total distance traveled by all the robots. Return the minimum total distance traveled by all the robots. The test cases are generated such that all the robots can be repaired. Note that

All robots move at the same speed. If two robots move in the same direction, they will never collide. If two robots move in opposite directions and they meet at some point, they do not collide. They cross each other. If a robot passes by a factory that reached its limits, it crosses it as if it does not exist. If the robot moved from a position x to a position y, the distance it moved is |y - x|.

Example 1:

Input: robot = [0,4,6], factory = [[2,2],[6,2]] Output: 4 Explanation: As shown in the figure: - The first robot at position 0 moves in the positive direction. It will be repaired at the first factory. - The second robot at position 4 moves in the negative direction. It will be repaired at the first factory. - The third robot at position 6 will be repaired at the second factory. It does not need to move. The limit of the first factory is 2, and it fixed 2 robots. The limit of the second factory is 2, and it fixed 1 robot. The total distance is |2 - 0| + |2 - 4| + |6 - 6| = 4. It can be shown that we cannot achieve a better total distance than 4.

Example 2:

Input: robot = [1,-1], factory = [[-2,1],[2,1]] Output: 2 Explanation: As shown in the figure: - The first robot at position 1 moves in the positive direction. It will be repaired at the second factory. - The second robot at position -1 moves in the negative direction. It will be repaired at the first factory. The limit of the first factory is 1, and it fixed 1 robot. The limit of the second factory is 1, and it fixed 1 robot. The total distance is |2 - 1| + |(-2) - (-1)| = 2. It can be shown that we cannot achieve a better total distance than 2.

Constraints:

1 <= robot.length, factory.length <= 100 factory[j].length == 2 -109 <= robot[i], positionj <= 109 0 <= limitj <= robot.length The input will be generated such that it is always possible to repair every robot.

==

Problem Number: 2109 URL: https://leetcode.com/problems/split-message-based-on-limit Title: 2468. Split Message Based on Limit Problem Description: You are given a string, message, and a positive integer, limit. You must split message into one or more parts based on limit. Each resulting part should have the suffix "<a/b>", where "b" is to be replaced with the total number of parts and "a" is to be replaced with the index of the part, starting from 1 and going up to b. Additionally, the length of each resulting part (including its suffix) should be equal to limit, except for the last part whose length can

be at most limit. The resulting parts should be formed such that when their suffixes are removed and they are all concatenated in order, they should be equal to message. Also, the result should contain as few parts as possible. Return the parts message would be split into as an array of strings. If it is impossible to split message as required, return an empty array. Example 1: Input: message = ”this is really a very awesome message”, limit = 9 Output: [”thi<1/14>”,”s i<2/14>”,”s r<3/14>”,”eal<4/14>”,”ly <5/14>”,”a v<6/14>”,”ery<7/14>”,” aw<8/14>”,”eso<9/14>”,”me<10/14>”,” m<11/14>”,”es<12/14>”,”sa<13/14>”
Explanation: The first 9 parts take 3 characters each from the beginning of message. The next 5 parts take 2 characters each to finish splitting message. In this example, each part, including the last, has length 9. It can be shown it is not possible to split message into less than 14 parts.

Example 2: Input: message = ”short message”, limit = 15 Output: [”short mess<1/2>”,”age<2/2>”] Explanation: Under the given constraints, the string can be split into two parts: - The first part comprises of the first 10 characters, and has a length 15. - The next part comprises of the last 3 characters, and has a length 8.

Constraints:

1 <= message.length <= 104 message consists only of lowercase English letters and ’ ’. 1 <= limit <= 104

==
Problem Number: 2110 URL: https://leetcode.com/problems/maximum-number-of-non-overlapping-palindrome-substrings Title: 2472. Maximum Number of Non-overlapping Palindrome Substrings Problem Description: You are given a string s and a positive integer k. Select a set of non-overlapping substrings from the string s that satisfy the following conditions:

The length of each substring is at least k. Each substring is a palindrome.

Return the maximum number of substrings in an optimal selection. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = ”abaccdbbd”, k = 3 Output: 2 Explanation: We can select the substrings underlined in s = ”abaccdbbd”. Both ”aba” and ”dbbd” are palindromes and have a length of at least k = 3. It can be shown that we cannot find a selection with more than two valid substrings.

Example 2: Input: s = ”adbcda”, k = 2 Output: 0 Explanation: There is no palindrome substring of length at least 2 in the string.

Constraints:

1 <= k <= s.length <= 2000 s consists of lowercase English letters.

==
Problem Number: 2111 URL: https://leetcode.com/problems/number-of-beautiful-partitions Title: 2478. Number of Beautiful Partitions Problem

Description: You are given a string s that consists of the digits '1' to '9' and two integers k and minLength. A partition of s is called beautiful if:

s is partitioned into k non-intersecting substrings. Each substring has a length of at least minLength. Each substring starts with a prime digit and ends with a non-prime digit. Prime digits are '2', '3', '5', and '7', and the rest of the digits are non-prime.

Return the number of beautiful partitions of s. Since the answer may be very large, return it modulo 109 + 7. A substring is a contiguous sequence of characters within a string. Example 1: Input: s = "23542185131", k = 3, minLength = 2 Output: 3 Explanation: There exists three ways to create a beautiful partition: "2354 | 218 | 5131" "2354 | 21851 | 31" "2354218 | 51 | 31"

Example 2: Input: s = "23542185131", k = 3, minLength = 3 Output: 1 Explanation: There exists one way to create a beautiful partition: "2354 | 218 | 5131".

Example 3: Input: s = "3312958", k = 3, minLength = 1 Output: 1 Explanation: There exists one way to create a beautiful partition: "331 | 29 | 58".

Constraints:

1 <= k, minLength <= s.length <= 1000 s consists of the digits '1' to '9'.

==
Problem Number: 2112 URL: https://leetcode.com/problems/count-palindromic-subsequences Title: 2484. Count Palindromic Subsequences Problem Description: Given a string of digits s, return the number of palindromic subsequences of s having length 5. Since the answer may be very large, return it modulo 109 + 7. Note:

A string is palindromic if it reads the same forward and backward. A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

Example 1: Input: s = "103301" Output: 2 Explanation: There are 6 possible subsequences of length 5: "10330","10331","10301","10301","13301","03301". Two of them (both equal to "10301") are palindromic.

Example 2: Input: s = "0000000" Output: 21 Explanation: All 21 subsequences are "00000", which is palindromic.

Example 3: Input: s = "9999900000" Output: 2 Explanation: The only two palindromic subsequences are "99999" and "00000".

Constraints:

1 <= s.length <= 104 s consists of digits.

==
Problem Number: 2113 URL: https://leetcode.com/problems/count-subarrays-with-median-k Title: 2488. Count Subarrays With Median K Problem

Description: You are given an array nums of size n consisting of distinct integers from 1 to n and a positive integer k. Return the number of non-empty subarrays in nums that have a median equal to k. Note:

The median of an array is the middle element after sorting the array in ascending order. If the array is of even length, the median is the left middle element.

For example, the median of [2,3,1,4] is 2, and the median of [8,4,3,5,1] is 4.

A subarray is a contiguous part of an array.

Example 1: Input: nums = [3,2,1,4,5], k = 4 Output: 3 Explanation: The subarrays that have a median equal to 4 are: [4], [4,5] and [1,4,5].

Example 2: Input: nums = [2,3,1], k = 3 Output: 1 Explanation: [3] is the only subarray that has a median equal to 3.

Constraints:

n == nums.length 1 <= n <= 105 1 <= nums[i], k <= n The integers in nums are distinct.

==
Problem Number: 2114 URL: https://leetcode.com/problems/divide-nodes-into-the-maximum-number-of-groups Title: 2493. Divide Nodes Into the Maximum Number of Groups Problem Description: You are given a positive integer n representing the number of nodes in an undirected graph. The nodes are labeled from 1 to n. You are also given a 2D integer array edges, where edges[i] = [ai, bi] indicates that there is a bidirectional edge between nodes ai and bi. Notice that the given graph may be disconnected. Divide the nodes of the graph into m groups (1-indexed) such that:

Each node in the graph belongs to exactly one group. For every pair of nodes in the graph that are connected by an edge [ai, bi], if ai belongs to the group with index x, and bi belongs to the group with index y, then |y - x| = 1.

Return the maximum number of groups (i.e., maximum m) into which you can divide the nodes. Return -1 if it is impossible to group the nodes with the given conditions. Example 1:

Input: n = 6, edges = [[1,2],[1,4],[1,5],[2,6],[2,3],[4,6]] Output: 4 Explanation: As shown in the image we: - Add node 5 to the first group. - Add node 1 to the second group. - Add nodes 2 and 4 to the third group. - Add nodes 3 and 6 to the fourth group. We can see that every edge is satisfied. It can be shown that that if we create a fifth group and move any node from the third or fourth group to it, at least on of the edges will not be satisfied.

Example 2: Input: n = 3, edges = [[1,2],[2,3],[3,1]] Output: -1 Explanation: If we add node 1 to the first group, node 2 to the second group, and node 3 to the third group to satisfy the first two edges, we can see that the third edge will not be satisfied. It can be shown that no grouping is possible.

Constraints:

1 <= n <= 500 1 <= edges.length <= 104 edges[i].length == 2 1 <= ai, bi <= n ai != bi There is at most one edge between any pair of vertices.

==
Problem Number: 2115 URL: https://leetcode.com/problems/minimum-total-cost-to-make-arrays-unequal Title: 2499. Minimum Total Cost to Make Arrays Unequal Problem Description: You are given two 0-indexed integer arrays nums1 and nums2, of equal length n. In one operation, you can swap the values of any two indices of nums1. The cost of this operation is the sum of the indices. Find the minimum total cost of performing the given operation any number of times such that nums1[i] != nums2[i] for all 0 <= i <= n - 1 after performing all the operations. Return the minimum total cost such that nums1 and nums2 satisfy the above condition. In case it is not possible, return -1.   Example 1: Input: nums1 = [1,2,3,4,5], nums2 = [1,2,3,4,5] Output: 10 Explanation: One of the ways we can perform the operations is: - Swap values at indices 0 and 3, incurring cost = 0 + 3 = 3. Now, nums1 = [4,2,3,1,5] - Swap values at indices 1 and 2, incurring cost = 1 + 2 = 3. Now, nums1 = [4,3,2,1,5]. - Swap values at indices 0 and 4, incurring cost = 0 + 4 = 4. Now, nums1 =[5,3,2,1,4]. We can see that for each index i, nums1[i] != nums2[i]. The cost required here is 10. Note that there are other ways to swap values, but it can be proven that it is not possible to obtain a cost less than 10.

Example 2: Input: nums1 = [2,2,2,1,3], nums2 = [1,2,2,3,3] Output: 10 Explanation: One of the ways we can perform the operations is: - Swap values at indices 2 and 3, incurring cost = 2 + 3 = 5. Now, nums1 = [2,2,1,2,3]. - Swap values at indices 1 and 4, incurring cost = 1 + 4 = 5. Now, nums1 = [2,3,1,2,2]. The total cost needed here is 10, which is the minimum possible.

Example 3: Input: nums1 = [1,2,2], nums2 = [1,2,2] Output: -1 Explanation: It can be shown that it is not possible to satisfy the given conditions irrespective of the number of operations we perform. Hence, we return -1.

Constraints:

n == nums1.length == nums2.length 1 <= n <= 105 1 <= nums1[i], nums2[i] <= n

==
Problem Number: 2116 URL: https://leetcode.com/problems/maximum-number-of-points-from-grid-queries Title: 2503. Maximum Number of Points From Grid Queries Problem Description: You are given an m x n integer matrix grid and an array queries of size k. Find an array answer of size k such that for each integer queries[i] you start in the top left cell of the matrix and repeat the following process:

If queries[i] is strictly greater than the value of the current cell that you are in, then you get one point if it is your first time visiting this cell, and you can move to any adjacent cell in all 4 directions: up, down, left, and right. Otherwise, you do not get any points, and you end this process.

After the process, answer[i] is the maximum number of points you can get. Note that for each query you are allowed to visit the same cell multiple times. Return the resulting array answer. Example 1:

Input: grid = [[1,2,3],[2,5,7],[3,5,1]], queries = [5,6,2] Output: [5,8,1] Explanation: The diagrams above show which cells we visit to get points for each query. Example 2:

Input: grid = [[5,2,1],[1,1,2]], queries = [3] Output: [0] Explanation: We can not get any points because the value of the top left cell is already greater than or equal to 3.

 Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 1000 4 <= m * n <= 105 k == queries.length 1 <= k <= 104 1 <= grid[i][j], queries[i] <= 106

==

Problem Number: 2117 URL: https://leetcode.com/problems/add-edges-to-make-degrees-of-all-nodes-even Title: 2508. Add Edges to Make Degrees of All Nodes Even Problem Description: There is an undirected graph consisting of n nodes numbered from 1 to n. You are given the integer n and a 2D array edges where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi. The graph can be disconnected. You can add at most two additional edges (possibly none) to this graph so that there are no repeated edges and no self-loops. Return true if it is possible to make the degree of each node in the graph even, otherwise return false. The degree of a node is the number of edges connected to it. Example 1:

Input: n = 5, edges = [[1,2],[2,3],[3,4],[4,2],[1,4],[2,5]] Output: true Explanation: The above diagram shows a valid way of adding an edge. Every node in the resulting graph is connected to an even number of edges.

Example 2:

Input: n = 4, edges = [[1,2],[3,4]] Output: true Explanation: The above diagram shows a valid way of adding two edges. Example 3:

Input: n = 4, edges = [[1,2],[1,3],[1,4]] Output: false Explanation: It is not possible to obtain a valid graph with adding at most 2 edges. Constraints:

3 <= n <= 105 2 <= edges.length <= 105 edges[i].length == 2 1 <= ai, bi <= n ai != bi There are no repeated edges.

==

Problem Number: 2118 URL: https://leetcode.com/problems/cycle-length-queries-in-a-tree Title: 2509. Cycle Length Queries in a Tree Problem Description: You are given an integer n. There is a complete binary tree with

2n - 1 nodes. The root of that tree is the node with the value 1, and every node with a value val in the range [1, 2n - 1 - 1] has two children where:

The left node has the value 2 * val, and The right node has the value 2 * val + 1.

You are also given a 2D integer array queries of length m, where queries[i] = [ai, bi]. For each query, solve the following problem:

Add an edge between the nodes with values ai and bi. Find the length of the cycle in the graph. Remove the added edge between nodes with values ai and bi.

Note that:

A cycle is a path that starts and ends at the same node, and each edge in the path is visited only once. The length of a cycle is the number of edges visited in the cycle. There could be multiple edges between two nodes in the tree after adding the edge of the query.

Return an array answer of length m where answer[i] is the answer to the ith query.   Example 1:

Input: n = 3, queries = [[5,3],[4,7],[2,3]] Output: [4,5,3] Explanation: The diagrams above show the tree of 23 - 1 nodes. Nodes colored in red describe the nodes in the cycle after adding the edge. - After adding the edge between nodes 3 and 5, the graph contains a cycle of nodes [5,2,1,3]. Thus answer to the first query is 4. We delete the added edge and process the next query. - After adding the edge between nodes 4 and 7, the graph contains a cycle of nodes [4,2,1,3,7]. Thus answer to the second query is 5. We delete the added edge and process the next query. - After adding the edge between nodes 2 and 3, the graph contains a cycle of nodes [2,1,3]. Thus answer to the third query is 3. We delete the added edge.

Example 2:

Input: n = 2, queries = [[1,2]] Output: [2] Explanation: The diagram above shows the tree of 22 - 1 nodes. Nodes colored in red describe the nodes in the cycle after adding the edge. - After adding the edge between nodes 1 and 2, the graph contains a cycle of nodes [2,1]. Thus answer for the first query is 2. We delete the added edge.

Constraints:

2 <= n <= 30 m == queries.length 1 <= m <= 105 queries[i].length == 2 1 <= ai, bi <= 2n - 1 ai != bi

==
Problem Number: 2119 URL: https://leetcode.com/problems/count-anagrams Title: 2514. Count Anagrams Problem Description: You are given a string s containing one or more words. Every consecutive pair of words is separated by

a single space ' '. A string t is an anagram of string s if the ith word of t is a permutation of the ith word of s.

For example, "acb dfe" is an anagram of "abc def", but "def cab" and "adc bef" are not.

Return the number of distinct anagrams of s. Since the answer may be very large, return it modulo 109 + 7. Example 1: Input: s = "too hot" Output: 18 Explanation: Some of the anagrams of the given string are "too hot", "oot hot", "oto toh", "too toh", and "too oht".

Example 2: Input: s = "aa" Output: 1 Explanation: There is only one anagram possible for the given string. Constraints:

1 <= s.length <= 105 s consists of lowercase English letters and spaces ' '. There is single space between consecutive words.

==
Problem Number: 2120 URL: https://leetcode.com/problems/number-of-great-partitions Title: 2518. Number of Great Partitions Problem Description: You are given an array nums consisting of positive integers and an integer k. Partition the array into two ordered groups such that each element is in exactly one group. A partition is called great if the sum of elements of each group is greater than or equal to k. Return the number of distinct great partitions. Since the answer may be too large, return it modulo 109 + 7. Two partitions are considered distinct if some element nums[i] is in different groups in the two partitions. Example 1: Input: nums = [1,2,3,4], k = 4 Output: 6 Explanation: The great partitions are: ([1,2,3], [4]), ([1,3], [2,4]), ([1,4], [2,3]), ([2,3], [1,4]), ([2,4], [1,3]) and ([4], [1,2,3]).

Example 2: Input: nums = [3,3,3], k = 4 Output: 0 Explanation: There are no great partitions for this array.

Example 3: Input: nums = [6,6], k = 2 Output: 2 Explanation: We can either put nums[0] in the first partition or in the second partition. The great partitions will be ([6], [6]) and ([6], [6]).

Constraints:

1 <= nums.length, k <= 1000 1 <= nums[i] <= 109

==
Problem Number: 2121 URL: https://leetcode.com/problems/maximize-the-minimum-powered-city Title: 2528. Maximize the Minimum Powered City Problem Description: You are given a 0-indexed integer array stations of length n, where stations[i] represents the number of power stations in the ith city. Each power station can provide power to every city in a fixed range. In other words, if the range is denoted by r, then a power station at city i can provide power to all cities j such that |i - j| <= r and 0 <= i, j <= n - 1.

Note that |x| denotes absolute value. For example, |7 - 5| = 2 and |3 - 10| = 7.

The power of a city is the total number of power stations it is being provided power from. The government has sanctioned building k more power stations, each of which can be built in any city, and have the same range as the pre-existing ones. Given the two integers r and k, return the maximum possible minimum power of a city, if the additional power stations are built optimally. Note that you can build the k power stations in multiple cities. Example 1: Input: stations = [1,2,4,5,0], r = 1, k = 2 Output: 5 Explanation: One of the optimal ways is to install both the power stations at city 1. So stations will become [1,4,4,5,0]. - City 0 is provided by 1 + 4 = 5 power stations. - City 1 is provided by 1 + 4 + 4 = 9 power stations. - City 2 is provided by 4 + 4 + 5 = 13 power stations. - City 3 is provided by 5 + 4 = 9 power stations. - City 4 is provided by 5 + 0 = 5 power stations. So the minimum power of a city is 5. Since it is not possible to obtain a larger power, we return 5.

Example 2: Input: stations = [4,4,4,4], r = 0, k = 3 Output: 4 Explanation: It can be proved that we cannot make the minimum power of a city greater than 4.

Constraints:

n == stations.length 1 <= n <= 105 0 <= stations[i] <= 105 0 <= r <= n - 1 0 <= k <= 109

==
Problem Number: 2122 URL: https://leetcode.com/problems/time-to-cross-a-bridge Title: 2532. Time to Cross a Bridge Problem Description: There are k workers who want to move n boxes from an old warehouse to a new one. You are given the two integers n and k, and a 2D integer array time of size k x 4 where time[i] = [leftToRighti, pickOldi, rightToLefti, putNewi]. The warehouses are separated by a river and connected by a bridge. The old warehouse is on the right bank of the river, and the new warehouse is on the left bank of the river. Initially, all k workers are waiting on the left side of the bridge. To move the boxes, the ith worker (0-indexed) can :

Cross the bridge from the left bank (new warehouse) to the right bank (old warehouse) in leftToRighti minutes. Pick a box from the old warehouse and return to the bridge in pickOldi minutes. Different workers can pick up their boxes simultaneously. Cross the bridge from the right bank (old warehouse) to the left bank (new warehouse) in rightToLefti minutes. Put the box in the new warehouse and return to the bridge in putNewi minutes. Different workers can put their boxes simultaneously.

A worker i is less efficient than a worker j if either condition is met:

leftToRighti + rightToLefti > leftToRightj + rightToLeftj leftToRighti + rightToLefti == leftToRightj + rightToLeftj and i > j

The following rules regulate the movement of the workers through the bridge :

If a worker x reaches the bridge while another worker y is crossing the bridge, x waits at their side of the bridge. If the bridge is free, the worker waiting on the right side of the bridge gets to cross the bridge. If more than one worker is waiting on the right side, the one with the lowest efficiency crosses first. If the bridge is free and no worker is waiting on the right side, and at least one box remains at the old warehouse, the worker on the left side of the river gets to cross the bridge. If more than one worker is waiting on the left side, the one with the lowest efficiency crosses first.

Return the instance of time at which the last worker reaches the left bank of the river after all n boxes have been put in the new warehouse.   Example 1: Input: n = 1, k = 3, time = [[1,1,2,1],[1,1,3,1],[1,1,4,1]] Output: 6 Explanation: From 0 to 1: worker 2 crosses the bridge from the left bank to the right bank. From 1 to 2: worker 2 picks up a box from the old warehouse. From 2 to 6: worker 2 crosses the bridge from the right bank to the left bank. From 6 to 7: worker 2 puts a box at the new warehouse. The whole process ends after 7 minutes. We return 6 because the problem asks for the instance of time at which the last worker reaches the left bank.

Example 2: Input: n = 3, k = 2, time = [[1,9,1,8],[10,10,10,10]] Output: 50 Explanation: From 0 to 10: worker 1 crosses the bridge from the left bank to the right bank. From 10 to 20: worker 1 picks up a box from the old warehouse. From 10 to 11: worker 0 crosses the bridge from the left bank to the right bank. From 11 to 20: worker 0 picks up a box from the old warehouse. From 20 to 30: worker 1 crosses the bridge from the right bank to the left bank. From 30 to 40: worker 1 puts a box at the new warehouse. From 30 to 31: worker 0 crosses the bridge from the right bank to the left bank. From 31 to 39: worker 0 puts a box at the new warehouse. From 39 to 40: worker 0 crosses the bridge from the left bank to the right bank. From 40 to 49: worker 0 picks up a box from the old warehouse. From 49 to 50: worker 0 crosses the bridge from the right bank to the left bank. From 50 to 58: worker 0 puts a box at the new warehouse. The whole process ends after 58 minutes. We return 50 because the problem asks for the instance of time at which the last worker reaches the left bank.

Constraints:

1 <= n, k <= 104 time.length == k time[i].length == 4 1 <= leftToRighti, pickOldi, rightToLefti, putNewi <= 1000

==
Problem Number: 2123 URL: https://leetcode.com/problems/difference-between-maximum-and-minimum-price-sum Title: 2538. Difference Between Maximum and Minimum Price Sum Problem Description: There exists an undirected and initially unrooted tree with n nodes indexed from 0 to n - 1. You are given the integer n and a 2D integer array edges of length n - 1, where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. Each node has an associated price. You are given an integer array price, where price[i] is the price of the ith node. The price sum of a given path is the sum of the prices of all nodes lying on that path. The tree can be rooted

at any node root of your choice. The incurred cost after choosing root is the difference between the maximum and minimum price sum amongst all paths starting at root. Return the maximum possible cost amongst all possible root choices. Example 1:

Input: n = 6, edges = [[0,1],[1,2],[1,3],[3,4],[3,5]], price = [9,8,7,6,10,5] Output: 24 Explanation: The diagram above denotes the tree after rooting it at node 2. The first part (colored in red) shows the path with the maximum price sum. The second part (colored in blue) shows the path with the minimum price sum. - The first path contains nodes [2,1,3,4]: the prices are [7,8,6,10], and the sum of the prices is 31. - The second path contains the node [2] with the price [7]. The difference between the maximum and minimum price sum is 24. It can be proved that 24 is the maximum cost.

Example 2:

Input: n = 3, edges = [[0,1],[1,2]], price = [1,1,1] Output: 2 Explanation: The diagram above denotes the tree after rooting it at node 0. The first part (colored in red) shows the path with the maximum price sum. The second part (colored in blue) shows the path with the minimum price sum. - The first path contains nodes [0,1,2]: the prices are [1,1,1], and the sum of the prices is 3. - The second path contains node [0] with a price [1]. The difference between the maximum and minimum price sum is 2. It can be proved that 2 is the maximum cost.

Constraints:

1 <= n <= 105 edges.length == n - 1 0 <= ai, bi <= n - 1 edges represents a valid tree. price.length == n 1 <= price[i] <= 105

==
Problem Number: 2124 URL: https://leetcode.com/problems/check-if-point-is-reachable Title: 2543. Check if Point Is Reachable Problem Description: There exists an infinitely large grid. You are currently at point (1, 1), and you need to reach the point (targetX, targetY) using a finite number of steps. In one step, you can move from point (x, y) to any one of the following points:

(x, y - x) (x - y, y) (2 * x, y) (x, 2 * y)

Given two integers targetX and targetY representing the X-coordinate and Y-coordinate of your final position, return true if you can reach the point from (1, 1) using some number of steps, and false otherwise. Example 1: Input: targetX = 6, targetY = 9 Output: false Explanation: It is impossible to reach (6,9) from (1,1) using any sequence of moves, so false is returned.

Example 2: Input: targetX = 4, targetY = 7 Output: true Explanation: You can follow the path (1,1) -> (1,2) -> (1,4) -> (1,8) -> (1,7) -> (2,7) -> (4,7).

Constraints:

1 <= targetX, targetY <= 109

==
Problem Number: 2125 URL: https://leetcode.com/problems/minimum-cost-to-split-an-array Title: 2547. Minimum Cost to Split an Array Problem Description: You are given an integer array nums and an integer k. Split the array into some number of non-empty subarrays. The cost of a split is the sum of the importance value of each subarray in the split. Let trimmed(subarray) be the version of the subarray where all numbers which appear only once are removed.

For example, trimmed([3,1,2,4,3,4]) = [3,4,3,4].

The importance value of a subarray is k + trimmed(subarray).length.

For example, if a subarray is [1,2,3,3,3,4,4], then trimmed([1,2,3,3,3,4,4]) = [3,3,3,4,4].The importance value of this subarray will be k + 5.

Return the minimum possible cost of a split of nums. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [1,2,1,2,1,3,3], k = 2 Output: 8 Explanation: We split nums to have two subarrays: [1,2], [1,2,1,3,3]. The importance value of [1,2] is 2 + (0) = 2. The importance value of [1,2,1,3,3] is 2 + (2 + 2) = 6. The cost of the split is 2 + 6 = 8. It can be shown that this is the minimum possible cost among all the possible splits.

Example 2: Input: nums = [1,2,1,2,1], k = 2 Output: 6 Explanation: We split nums to have two subarrays: [1,2], [1,2,1]. The importance value of [1,2] is 2 + (0) = 2. The importance value of [1,2,1] is 2 + (2) = 4. The cost of the split is 2 + 4 = 6. It can be shown that this is the minimum possible cost among all the possible splits.

Example 3: Input: nums = [1,2,1,2,1], k = 5 Output: 10 Explanation: We split nums to have one subarray: [1,2,1,2,1]. The importance value of [1,2,1,2,1] is 5 + (3 + 2) = 10. The cost of the split is 10. It can be shown that this is the minimum possible cost among all the possible splits.

Constraints:

1 <= nums.length <= 1000 0 <= nums[i] < nums.length 1 <= k <= 109

==
Problem Number: 2126 URL: https://leetcode.com/problems/put-marbles-in-bags Title: 2551. Put Marbles in Bags Problem Description: You have k bags. You are given a 0-indexed integer array weights where weights[i] is the weight of the ith marble. You are also given the integer k. Divide the marbles into the k bags according to the following rules:

No bag is empty. If the ith marble and jth marble are in a bag, then all marbles with an index between the ith and jth indices should also be in that same bag.

If a bag consists of all the marbles with an index from i to j inclusively, then the cost of the bag is weights[i] + weights[j].

The score after distributing the marbles is the sum of the costs of all the k bags. Return the difference between the maximum and minimum scores among marble distributions.   Example 1: Input: weights = [1,3,5,1], k = 2 Output: 4 Explanation: The distribution [1],[3,5,1] results in the minimal score of (1+1) + (3+1) = 6. The distribution [1,3],[5,1], results in the maximal score of (1+3) + (5+1) = 10. Thus, we return their difference 10 - 6 = 4.

Example 2: Input: weights = [1, 3], k = 2 Output: 0 Explanation: The only distribution possible is [1],[3]. Since both the maximal and minimal score are the same, we return 0.

Constraints:

1 <= k <= weights.length <= 105 1 <= weights[i] <= 109

==
Problem Number: 2127 URL: https://leetcode.com/problems/count-increasing-quadruplets Title: 2552. Count Increasing Quadruplets Problem Description: Given a 0-indexed integer array nums of size n containing all numbers from 1 to n, return the number of increasing quadruplets. A quadruplet (i, j, k, l) is increasing if:

0 <= i < j < k < l < n, and nums[i] < nums[k] < nums[j] < nums[l].

Example 1: Input: nums = [1,3,2,4,5] Output: 2 Explanation: - When i = 0, j = 1, k = 2, and l = 3, nums[i] < nums[k] < nums[j] < nums[l]. - When i = 0, j = 1, k = 2, and l = 4, nums[i] < nums[k] < nums[j] < nums[l]. There are no other quadruplets, so we return 2.

Example 2: Input: nums = [1,2,3,4] Output: 0 Explanation: There exists only one quadruplet with i = 0, j = 1, k = 2, l = 3, but since nums[j] < nums[k], we return 0.

Constraints:

4 <= nums.length <= 4000 1 <= nums[i] <= nums.length All the integers of nums are unique. nums is a permutation.

==
Problem Number: 2128 URL: https://leetcode.com/problems/rearranging-fruits Title: 2561. Rearranging Fruits Problem Description: You have two fruit baskets containing n fruits each. You are given two 0-indexed integer arrays basket1 and basket2 representing the cost of fruit in each basket. You want to make both baskets equal. To do so, you can use the following operation as many times as you want:

Chose two indices i and j, and swap the ith fruit of basket1 with the jth fruit of basket2. The cost of the swap is min(basket1[i],basket2[j]).

Two baskets are considered equal if sorting them according to the fruit cost makes them exactly the same baskets. Return the minimum cost to make both the baskets equal or -1 if impossible.   Example 1: Input: basket1 = [4,2,2,2], basket2 = [1,4,1,2] Output: 1 Explanation: Swap index 1 of basket1 with index 0 of basket2, which has cost 1. Now basket1 = [4,1,2,2] and basket2 = [2,4,1,2]. Rearranging both the arrays makes them equal.

Example 2: Input: basket1 = [2,3,4,1], basket2 = [3,2,5,1] Output: -1 Explanation: It can be shown that it is impossible to make both the baskets equal.

Constraints:

basket1.length == basket2.length 1 <= basket1.length <= 105 1 <= basket1[i],basket2[i] <= 109

==
Problem Number: 2129 URL: https://leetcode.com/problems/subsequence-with-the-minimum-score Title: 2565. Subsequence With the Minimum Score Problem Description: You are given two strings s and t. You are allowed to remove any number of characters from the string t. The score of the string is 0 if no characters are removed from the string t, otherwise:

Let left be the minimum index among all removed characters. Let right be the maximum index among all removed characters.

Then the score of the string is right - left + 1. Return the minimum possible score to make t a subsequence of s. A subsequence of a string is a new string that is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (i.e., "ace" is a subsequence of "abcde" while "aec" is not).   Example 1: Input: s = "abacaba", t = "bzaa" Output: 1 Explanation: In this example, we remove the character "z" at index 1 (0-indexed). The string t becomes "baa" which is a subsequence of the string "abacaba" and the score is 1 - 1 + 1 = 1. It can be proven that 1 is the minimum score that we can achieve.

Example 2: Input: s = "cde", t = "xyz" Output: 3 Explanation: In this example, we remove characters "x", "y" and "z" at indices 0, 1, and 2 (0-indexed). The string t becomes "" which is a subsequence of the string "cde" and the score is 2 - 0 + 1 = 3. It can be proven that 3 is the minimum score that we can achieve.

Constraints:

1 <= s.length, t.length <= 105 s and t consist of only lowercase English letters.

==
Problem Number: 2130 URL: https://leetcode.com/problems/handling-sum-queries-after-update Title: 2569. Handling Sum Queries After Update Problem Description: You are given two 0-indexed arrays nums1 and nums2 and a 2D array queries of queries. There are three types of queries:

For a query of type 1, queries[i] = [1, l, r]. Flip the values from 0 to 1 and from

1 to 0 in nums1 from index l to index r. Both l and r are 0-indexed. For a query of type 2, queries[i] = [2, p, 0]. For every index 0 <= i < n, set nums2[i] = nums2[i] + nums1[i] * p. For a query of type 3, queries[i] = [3, 0, 0]. Find the sum of the elements in nums2.

Return an array containing all the answers to the third type queries. Example 1: Input: nums1 = [1,0,1], nums2 = [0,0,0], queries = [[1,1,1],[2,1,0],[3,0,0]] Output: [3] Explanation: After the first query nums1 becomes [1,1,1]. After the second query, nums2 becomes [1,1,1], so the answer to the third query is 3. Thus, [3] is returned.

Example 2: Input: nums1 = [1], nums2 = [5], queries = [[2,0,0],[3,0,0]] Output: [5] Explanation: After the first query, nums2 remains [5], so the answer to the second query is 5. Thus, [5] is returned.

Constraints:

1 <= nums1.length,nums2.length <= 105 nums1.length = nums2.length 1 <= queries.length <= 105 queries[i].length = 3 0 <= l <= r <= nums1.length - 1 0 <= p <= 106 0 <= nums1[i] <= 1 0 <= nums2[i] <= 109

==
Problem Number: 2131 URL: https://leetcode.com/problems/find-the-string-with-lcp Title: 2573. Find the String with LCP Problem Description: We define the lcp matrix of any 0-indexed string word of n lowercase English letters as an n x n grid such that:

lcp[i][j] is equal to the length of the longest common prefix between the substrings word[i,n-1] and word[j,n-1].

Given an n x n matrix lcp, return the alphabetically smallest string word that corresponds to lcp. If there is no such string, return an empty string. A string a is lexicographically smaller than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b. For example, "aabd" is lexicographically smaller than "aaca" because the first position they differ is at the third letter, and 'b' comes before 'c'. Example 1: Input: lcp = [[4,0,2,0],[0,3,0,1],[2,0,2,0],[0,1,0,1]] Output: "abab" Explanation: lcp corresponds to any 4 letter string with two alternating letters. The lexicographically smallest of them is "abab".

Example 2: Input: lcp = [[4,3,2,1],[3,3,2,1],[2,2,2,1],[1,1,1,1]] Output: "aaaa" Explanation: lcp corresponds to any 4 letter string with a single distinct letter. The lexicographically smallest of them is "aaaa".

Example 3: Input: lcp = [[4,3,2,1],[3,3,2,1],[2,2,2,1],[1,1,1,3]] Output: "" Explanation: lcp[3][3] cannot be equal to 3 since word[3,...,3] consists of only a single letter; Thus, no answer exists.

Constraints:

1 <= n == lcp.length == lcp[i].length <= 1000 0 <= lcp[i][j] <= n

== Problem Number: 2132 URL: https://leetcode.com/problems/minimum-time-to-visit-a-cell-in-a-grid Title: 2577. Minimum Time to Visit a Cell In a Grid Problem Description: You are given a m x n matrix grid consisting of non-negative integers where grid[row][col] represents the minimum time required to be able to visit the cell (row, col), which means you can visit the cell (row, col) only when the time you visit it is greater than or equal to grid[row][col]. You are standing in the top-left cell of the matrix in the 0th second, and you must move to any adjacent cell in the four directions: up, down, left, and right. Each move you make takes 1 second. Return the minimum time required in which you can visit the bottom-right cell of the matrix. If you cannot visit the bottom-right cell, then return -1.   Example 1:

Input: grid = [[0,1,3,2],[5,1,2,5],[4,3,8,6]] Output: 7 Explanation: One of the paths that we can take is the following: - at t = 0, we are on the cell (0,0). - at t = 1, we move to the cell (0,1). It is possible because grid[0][1] <= 1. - at t = 2, we move to the cell (1,1). It is possible because grid[1][1] <= 2. - at t = 3, we move to the cell (1,2). It is possible because grid[1][2] <= 3. - at t = 4, we move to the cell (1,1). It is possible because grid[1][1] <= 4. - at t = 5, we move to the cell (1,2). It is possible because grid[1][2] <= 5. - at t = 6, we move to the cell (1,3). It is possible because grid[1][3] <= 6. - at t = 7, we move to the cell (2,3). It is possible because grid[2][3] <= 7. The final time is 7. It can be shown that it is the minimum time possible.

Example 2:

Input: grid = [[0,2,4],[3,2,1],[1,0,4]] Output: -1 Explanation: There is no path from the top left to the bottom-right cell.

Constraints:

m == grid.length n == grid[i].length 2 <= m, n <= 1000 4 <= m * n <= 105 0 <= grid[i][j] <= 105 grid[0][0] == 0

== Problem Number: 2133 URL: https://leetcode.com/problems/count-number-of-possible-root-nodes Title: 2581. Count Number of Possible Root Nodes Problem Description: Alice has an undirected tree with n nodes labeled from 0 to n - 1. The tree is represented as a 2D integer array edges of length n - 1 where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. Alice wants Bob to find the root of the tree. She allows Bob to make several guesses about her tree. In one guess, he does the following:

Chooses two distinct integers u and v such that there exists an edge [u, v] in the tree. He tells Alice that u is the parent of v in the tree.

Bob's guesses are represented by a 2D integer array guesses where guesses[j] = [uj, vj] indicates Bob guessed uj to be the parent of vj. Alice being lazy, does not reply to each of Bob's guesses, but just says that at least k of his guesses are true. Given the 2D integer arrays edges, guesses and the integer k, return the number of possible nodes that can be the root of Alice's tree. If there is no such tree, return 0. Example 1:

Input: edges = [[0,1],[1,2],[1,3],[4,2]], guesses = [[1,3],[0,1],[1,0],[2,4]], k = 3 Output: 3 Explanation: Root = 0, correct guesses = [1,3], [0,1], [2,4] Root = 1, correct guesses = [1,3], [1,0], [2,4] Root = 2, correct guesses = [1,3], [1,0], [2,4] Root = 3, correct guesses = [1,0], [2,4] Root = 4, correct guesses = [1,3], [1,0] Considering 0, 1, or 2 as root node leads to 3 correct guesses.

Example 2:

Input: edges = [[0,1],[1,2],[2,3],[3,4]], guesses = [[1,0],[3,4],[2,1],[3,2]], k = 1 Output: 5 Explanation: Root = 0, correct guesses = [3,4] Root = 1, correct guesses = [1,0], [3,4] Root = 2, correct guesses = [1,0], [2,1], [3,4] Root = 3, correct guesses = [1,0], [2,1], [3,2], [3,4] Root = 4, correct guesses = [1,0], [2,1], [3,2] Considering any node as root will give at least 1 correct guess.

Constraints:

edges.length == n - 1 2 <= n <= 105 1 <= guesses.length <= 105 0 <= ai, bi, uj, vj <= n - 1 ai != bi uj != vj edges represents a valid tree. guesses[j] is an edge of the tree. guesses is unique. 0 <= k <= guesses.length

==

Problem Number: 2134 URL: https://leetcode.com/problems/split-the-array-to-make-coprime-products Title: 2584. Split the Array to Make Coprime Products Problem Description: You are given a 0-indexed integer array nums of length n. A split at an index i where 0 <= i <= n - 2 is called valid if the product of the first i + 1 elements and the product of the remaining elements are coprime.

For example, if nums = [2, 3, 3], then a split at the index i = 0 is valid because 2 and 9 are coprime, while a split at the index i = 1 is not valid because 6 and 3 are not coprime. A split at the index i = 2 is not valid because i == n - 1.

Return the smallest index i at which the array can be split validly or -1 if there is no such split. Two values val1 and val2 are coprime if gcd(val1, val2) == 1 where gcd(val1, val2) is the greatest common divisor of val1 and val2. Example 1:

Input: nums = [4,7,8,15,3,5] Output: 2 Explanation: The table above shows the values of the product of the first i + 1 elements, the remaining elements, and their gcd at each index i. The only valid split is at index 2.

Example 2:

Input: nums = [4,7,15,8,3,5] Output: -1 Explanation: The table above shows the values of the product of the first i + 1 elements, the remaining elements, and their gcd at each index i. There is no valid split.

Constraints:

n == nums.length 1 <= n <= 104 1 <= nums[i] <= 106

==
Problem Number: 2135 URL: https://leetcode.com/problems/number-of-ways-to-earn-points Title: 2585. Number of Ways to Earn Points Problem Description: There is a test that has n types of questions. You are given an integer target and a 0-indexed 2D integer array types where types[i] = [counti, marksi] indicates that there are counti questions of the ith type, and each one of them is worth marksi points.

Return the number of ways you can earn exactly target points in the exam. Since the answer may be too large, return it modulo 109 + 7. Note that questions of the same type are indistinguishable.

For example, if there are 3 questions of the same type, then solving the 1st and 2nd questions is the same as solving the 1st and 3rd questions, or the 2nd and 3rd questions.

Example 1: Input: target = 6, types = [[6,1],[3,2],[2,3]] Output: 7 Explanation: You can earn 6 points in one of the seven ways: - Solve 6 questions of the 0th type: 1 + 1 + 1 + 1 + 1 + 1 = 6 - Solve 4 questions of the 0th type and 1 question of the 1st type: 1 + 1 + 1 + 1 + 2 = 6 - Solve 2 questions of the 0th type and 2 questions of the 1st type: 1 + 1 + 2 + 2 = 6 - Solve 3 questions of the 0th type and 1 question of the 2nd type: 1 + 1 + 1 + 3 = 6 - Solve 1 question of the 0th type, 1 question of the 1st type and 1 question of the 2nd type: 1 + 2 + 3 = 6 - Solve 3 questions of the 1st type: 2 + 2 + 2 = 6 - Solve 2 questions of the 2nd type: 3 + 3 = 6

Example 2: Input: target = 5, types = [[50,1],[50,2],[50,5]] Output: 4 Explanation: You can earn 5 points in one of the four ways: - Solve 5 questions of the 0th type: 1 + 1 + 1 + 1 + 1 = 5 - Solve 3 questions of the 0th type and 1 question of the 1st type: 1 + 1 + 1 + 2 = 5 - Solve 1 questions of the 0th type and 2 questions of the 1st type: 1 + 2 + 2 = 5 - Solve 1 question of the 2nd type: 5

Example 3: Input: target = 18, types = [[6,1],[3,2],[2,3]] Output: 1 Explanation: You can only earn 18 points by answering all questions.

Constraints:

1 <= target <= 1000 n == types.length 1 <= n <= 50 types[i].length == 2 1 <= counti, marksi <= 50

==
Problem Number: 2136 URL: https://leetcode.com/problems/minimum-time-to-complete-all-tasks Title: 2589. Minimum Time to Complete All Tasks

Problem Description: There is a computer that can run an unlimited number of tasks at the same time. You are given a 2D integer array tasks where tasks[i] = [starti, endi, durationi] indicates that the ith task should run for a total of durationi seconds (not necessarily continuous) within the inclusive time range [starti, endi]. You may turn on the computer only when it needs to run a task. You can also turn it off if it is idle. Return the minimum time during which the computer should be turned on to complete all tasks.   Example 1: Input: tasks = [[2,3,1],[4,5,1],[1,5,2]] Output: 2 Explanation: - The first task can be run in the inclusive time range [2, 2]. - The second task can be run in the inclusive time range [5, 5]. - The third task can be run in the two inclusive time ranges [2, 2] and [5, 5]. The computer will be on for a total of 2 seconds.

Example 2: Input: tasks = [[1,3,2],[2,5,3],[5,6,2]] Output: 4 Explanation: - The first task can be run in the inclusive time range [2, 3]. - The second task can be run in the inclusive time ranges [2, 3] and [5, 5]. - The third task can be run in the two inclusive time range [5, 6]. The computer will be on for a total of 4 seconds.

Constraints:

1 <= tasks.length <= 2000 tasks[i].length == 3 1 <= starti, endi <= 2000 1 <= durationi <= endi - starti + 1

==
Problem Number: 2137 URL: https://leetcode.com/problems/collect-coins-in-a-tree Title: 2603. Collect Coins in a Tree Problem Description: There exists an undirected and unrooted tree with n nodes indexed from 0 to n - 1. You are given an integer n and a 2D integer array edges of length n - 1, where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. You are also given an array coins of size n where coins[i] can be either 0 or 1, where 1 indicates the presence of a coin in the vertex i. Initially, you choose to start at any vertex in the tree. Then, you can perform the following operations any number of times:

Collect all the coins that are at a distance of at most 2 from the current vertex, or Move to any adjacent vertex in the tree.

Find the minimum number of edges you need to go through to collect all the coins and go back to the initial vertex. Note that if you pass an edge several times, you need to count it into the answer several times.   Example 1:

Input: coins = [1,0,0,0,0,1], edges = [[0,1],[1,2],[2,3],[3,4],[4,5]] Output: 2 Explanation: Start at vertex 2, collect the coin at vertex 0, move to vertex 3, collect the coin at vertex 5 then move back to vertex 2.

Example 2:

Input: coins = [0,0,0,1,1,0,0,1], edges = [[0,1],[0,2],[1,3],[1,4],[2,5],[5,6],[5,7]] Output: 2 Explanation: Start at vertex 0, collect the coins at vertices 4 and 3, move to vertex 2, collect the coin at vertex 7, then move back to vertex 0.

Constraints:

n == coins.length 1 <= n <= 3 * 104 0 <= coins[i] <= 1 edges.length == n - 1 edges[i].length == 2 0 <= ai, bi < n ai != bi edges represents a valid tree.

==
Problem Number: 2138 URL: https://leetcode.com/problems/shortest-cycle-in-a-graph Title: 2608. Shortest Cycle in a Graph Problem Description: There is a bi-directional graph with n vertices, where each vertex is labeled from 0 to n - 1. The edges in the graph are represented by a given 2D integer array edges, where edges[i] = [ui, vi] denotes an edge between vertex ui and vertex vi. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself. Return the length of the shortest cycle in the graph. If no cycle exists, return -1. A cycle is a path that starts and ends at the same node, and each edge in the path is used only once.   Example 1:

Input: n = 7, edges = [[0,1],[1,2],[2,0],[3,4],[4,5],[5,6],[6,3]] Output: 3 Explanation: The cycle with the smallest length is : 0 -> 1 -> 2 -> 0

Example 2:

Input: n = 4, edges = [[0,1],[0,2]] Output: -1 Explanation: There are no cycles in this graph.

Constraints:

2 <= n <= 1000 1 <= edges.length <= 1000 edges[i].length == 2 0 <= ui, vi < n ui != vi There are no repeated edges.

==
Problem Number: 2139 URL: https://leetcode.com/problems/minimum-reverse-operations Title: 2612. Minimum Reverse Operations Problem Description: You are given an integer n and an integer p in the range [0, n - 1]. Representing a 0-indexed array arr of length n where all positions are set to 0's, except position p which is set to 1. You are also given an integer array banned containing some positions from the array. For the ith position in banned, arr[banned[i]] = 0, and banned[i] != p. You can perform multiple operations on arr. In an operation, you can choose a subarray with size k and reverse the subarray. However, the 1 in arr should never go to any of the positions in banned. In other words, after each operation arr[banned[i]] remains 0. Return an array ans where for each i from [0, n - 1], ans[i] is the minimum number of reverse operations needed to bring the 1 to position i in arr, or -1 if it is impossible.

A subarray is a contiguous non-empty sequence of elements within an array. The values of ans[i] are independent for all i's. The reverse of an array is an array containing the values in reverse order.

Example 1: Input: n = 4, p = 0, banned = [1,2], k = 4 Output: [0,-1,-1,1] Explanation: In this case k = 4 so there is only one possible reverse operation we can perform, which is reversing the whole array. Initially, 1 is placed at position 0 so the amount of operations we need for position 0 is 0. We can never place a 1 on the banned positions, so the answer for positions 1 and 2 is -1. Finally, with one reverse operation we can bring the 1 to index 3, so the answer for position 3 is 1.

Example 2: Input: n = 5, p = 0, banned = [2,4], k = 3 Output: [0,-1,-1,-1,-1] Explanation: In this case the 1 is initially at position 0, so the answer for that position is 0. We can perform reverse operations of size 3. The 1 is currently located at position 0, so we need to reverse the subarray [0, 2] for it to leave that position, but reversing that subarray makes position 2 have a 1, which shouldn't happen. So, we can't move the 1 from position 0, making the result for all the other positions -1.

Example 3: Input: n = 4, p = 2, banned = [0,1,3], k = 1 Output: [-1,-1,0,-1] Explanation: In this case we can only perform reverse operations of size 1. So the 1 never changes its position.

Constraints:

1 <= n <= 105 0 <= p <= n - 1 0 <= banned.length <= n - 1 0 <= banned[i] <= n - 1 1 <= k <= n banned[i] != p all values in banned are unique

==

Problem Number: 2140 URL: https://leetcode.com/problems/minimum-number-of-visited-cells-in-a-grid Title: 2617. Minimum Number of Visited Cells in a Grid Problem Description: You are given a 0-indexed m x n integer matrix grid. Your initial position is at the top-left cell (0, 0). Starting from the cell (i, j), you can move to one of the following cells:

Cells (i, k) with j < k <= grid[i][j] + j (rightward movement), or Cells (k, j) with i < k <= grid[i][j] + i (downward movement).

Return the minimum number of cells you need to visit to reach the bottom-right cell (m - 1, n - 1). If there is no valid path, return -1. Example 1:

Input: grid = [[3,4,2,1],[4,2,3,1],[2,1,0,0],[2,4,0,0]] Output: 4 Explanation: The image above shows one of the paths that visits exactly 4 cells.

Example 2:

Input: grid = [[3,4,2,1],[4,2,1,1],[2,1,1,0],[3,4,1,0]] Output: 3 Explanation: The image above shows one of the paths that visits exactly 3 cells.

Example 3:

Input: grid = [[2,1,0],[1,0,0]] Output: -1 Explanation: It can be proven that no path exists.

Constraints:

m == grid.length n == grid[i].length 1 <= m, n <= 105 1 <= m * n <= 105 0 <= grid[i][j] < m * n grid[m - 1][n - 1] == 0

==
Problem Number: 2141 URL: https://leetcode.com/problems/design-graph-with-shortest-path-calculator Title: 2642. Design Graph With Shortest Path Calculator Problem Description: There is a directed weighted graph that consists of n nodes numbered from 0 to n - 1. The edges of the graph are initially represented by the given array edges where edges[i] = [fromi, toi, edgeCosti] meaning that there is an edge from fromi to toi with the cost edgeCosti. Implement the Graph class:

Graph(int n, int[][] edges) initializes the object with n nodes and the given edges. addEdge(int[] edge) adds an edge to the list of edges where edge = [from, to, edgeCost]. It is guaranteed that there is no edge between the two nodes before adding this one. int shortestPath(int node1, int node2) returns the minimum cost of a path from node1 to node2. If no path exists, return -1. The cost of a path is the sum of the costs of the edges in the path.

Example 1:

Input ["Graph", "shortestPath", "shortestPath", "addEdge", "shortestPath"] [[4, [[0, 2, 5], [0, 1, 2], [1, 2, 1], [3, 0, 3]]], [3, 2], [0, 3], [[1, 3, 4]], [0, 3]] Output [null, 6, -1, null, 6]

Explanation Graph g = new Graph(4, [[0, 2, 5], [0, 1, 2], [1, 2, 1], [3, 0, 3]]); g.shortestPath(3, 2); // return 6. The shortest path from 3 to 2 in the first diagram above is 3 -> 0 -> 1 -> 2 with a total cost of 3 + 2 + 1 = 6. g.shortestPath(0, 3); // return -1. There is no path from 0 to 3. g.addEdge([1, 3, 4]); // We add an edge from node 1 to node 3, and we get the second diagram above. g.shortestPath(0, 3); // return 6. The shortest path from 0 to 3 now is 0 -> 1 -> 3 with a total cost of 2 + 4 = 6.

Constraints:

1 <= n <= 100 0 <= edges.length <= n * (n - 1) edges[i].length == edge.length == 3 0 <= fromi, toi, from, to, node1, node2 <= n - 1 1 <= edgeCosti, edgeCost <= 106 There are no repeated edges and no self-loops in the graph at any point. At most 100 calls will be made for addEdge. At most 100 calls will be made for shortestPath.

==
Problem Number: 2142 URL: https://leetcode.com/problems/minimize-the-total-price-of-the-trips Title: 2646. Minimize the Total Price of the Trips Problem Description: There exists an undirected and unrooted tree with n nodes indexed from 0 to n - 1. You are given the integer n and a 2D integer array edges of length n - 1, where edges[i] = [ai, bi] indicates that there is an edge between nodes ai and bi in the tree. Each node has an associated price. You are given an integer array price, where price[i] is the price of the ith node. The price sum of a given path is the sum of the prices of all nodes lying on that path. Additionally, you are given a 2D integer array trips, where trips[i] = [starti, endi] indicates that you start the ith trip from the node starti and

travel to the node endi by any path you like. Before performing your first trip, you can choose some non-adjacent nodes and halve the prices. Return the minimum total price sum to perform all the given trips.   Example 1:

Input: n = 4, edges = [[0,1],[1,2],[1,3]], price = [2,2,10,6], trips = [[0,3],[2,1],[2,3]] Output: 23 Explanation: The diagram above denotes the tree after rooting it at node 2. The first part shows the initial tree and the second part shows the tree after choosing nodes 0, 2, and 3, and making their price half. For the 1st trip, we choose path [0,1,3]. The price sum of that path is 1 + 2 + 3 = 6. For the 2nd trip, we choose path [2,1]. The price sum of that path is 2 + 5 = 7. For the 3rd trip, we choose path [2,1,3]. The price sum of that path is 5 + 2 + 3 = 10. The total price sum of all trips is 6 + 7 + 10 = 23. It can be proven, that 23 is the minimum answer that we can achieve.

Example 2:

Input: n = 2, edges = [[0,1]], price = [2,2], trips = [[0,0]] Output: 1 Explanation: The diagram above denotes the tree after rooting it at node 0. The first part shows the initial tree and the second part shows the tree after choosing node 0, and making its price half. For the 1st trip, we choose path [0]. The price sum of that path is 1. The total price sum of all trips is 1. It can be proven, that 1 is the minimum answer that we can achieve.

Constraints:

1 <= n <= 50 edges.length == n - 1 0 <= ai, bi <= n - 1 edges represents a valid tree. price.length == n price[i] is an even integer. 1 <= price[i] <= 1000 1 <= trips.length <= 100 0 <= starti, endi <= n - 1

==
Problem Number: 2143 URL: https://leetcode.com/problems/make-array-empty Title: 2659. Make Array Empty Problem Description: You are given an integer array nums containing distinct numbers, and you can perform the following operations until the array is empty:

If the first element has the smallest value, remove it Otherwise, put the first element at the end of the array.

Return an integer denoting the number of operations it takes to make nums empty.   Example 1: Input: nums = [3,4,-1] Output: 5

OperationArray1[4, -1, 3]2[-1, 3, 4]3[3, 4]4[4]5[] Example 2: Input: nums = [1,2,4,3] Output: 5

OperationArray1[2, 4, 3]2[4, 3]3[3, 4]4[4]5[] Example 3: Input: nums = [1,2,3] Output: 3

OperationArray1[2, 3]2[3]3[]   Constraints:

1 <= nums.length <= 105 -109 <= nums[i] <= 109 All values in nums are distinct.

==
Problem Number: 2144 URL: https://leetcode.com/problems/lexicographically-smallest-beautiful-string Title: 2663. Lexicographically Smallest Beautiful String Problem Description: A string is beautiful if:

It consists of the first k letters of the English lowercase alphabet. It does not contain any substring of length 2 or more which is a palindrome.

You are given a beautiful string s of length n and a positive integer k. Return the lexicographically smallest string of length n, which is larger than s and is beautiful. If there is no such string, return an empty string. A string a is lexicographically larger than a string b (of the same length) if in the first position where a and b differ, a has a character strictly larger than the corresponding character in b.

For example, "abcd" is lexicographically larger than "abcc" because the first position they differ is at the fourth character, and d is greater than c.

Example 1: Input: s = "abcz", k = 26 Output: "abda" Explanation: The string "abda" is beautiful and lexicographically larger than the string "abcz". It can be proven that there is no string that is lexicographically larger than the string "abcz", beautiful, and lexicographically smaller than the string "abda".

Example 2: Input: s = "dc", k = 4 Output: "" Explanation: It can be proven that there is no string that is lexicographically larger than the string "dc" and is beautiful.

Constraints:

1 <= n == s.length <= 105 4 <= k <= 26 s is a beautiful string.

==
Problem Number: 2145 URL: https://leetcode.com/problems/power-of-heroes Title: 2681. Power of Heroes Problem Description: You are given a 0-indexed integer array nums representing the strength of some heroes. The power of a group of heroes is defined as follows:

Let i0, i1, ... ,ik be the indices of the heroes in a group. Then, the power of this group is max(nums[i0], nums[i1], ... ,nums[ik])2 * min(nums[i0], nums[i1], ... ,nums[ik]).

Return the sum of the power of all non-empty groups of heroes possible. Since the sum could be very large, return it modulo 109 + 7. Example 1: Input: nums = [2,1,4] Output: 141 Explanation: 1st group: [2] has power = 22 * 2 = 8. 2nd group: [1] has power = 12 * 1 = 1. 3rd group: [4] has power = 42 * 4 = 64. 4th group: [2,1] has power = 22 * 1 = 4. 5th group: [2,4] has power = 42 * 2 = 32. 6th group: [1,4] has power = 42 * 1 = 16. 7th group: [2,1,4] has power = 42 * 1 = 16. The sum of powers of all groups is 8 + 1 + 64 + 4 + 32 + 16 + 16 = 141.

Example 2: Input: nums = [1,1,1] Output: 7 Explanation: A total of 7 groups are possible, and the power of each group will be 1. Therefore, the sum of the powers of all groups is 7.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 109

==
Problem Number: 2146 URL: https://leetcode.com/problems/modify-graph-edge-weights Title: 2699. Modify Graph Edge Weights Problem Description: You are given an undirected weighted connected graph containing n nodes labeled from 0 to n - 1, and an integer array edges where edges[i] = [ai, bi, wi] indicates that there is an edge between nodes ai and bi with weight wi. Some edges have a weight of -1 (wi = -1), while others have a positive weight (wi > 0). Your task is to modify all edges with a weight of -1 by assigning them positive integer values in the range [1, 2 * 109] so that the shortest distance between the nodes source and destination becomes equal to an integer target. If there are multiple modifications that make the shortest distance between source and destination equal to target, any of them will be considered correct. Return an array containing all edges (even unmodified ones) in any order if it is possible to make the shortest distance from source to destination equal to target, or an empty array if it's impossible. Note: You are not allowed to modify the weights of edges with initial positive weights. Example 1:

Input: n = 5, edges = [[4,1,-1],[2,0,-1],[0,3,-1],[4,3,-1]], source = 0, destination = 1, target = 5 Output: [[4,1,1],[2,0,1],[0,3,3],[4,3,1]] Explanation: The graph above shows a possible modification to the edges, making the distance from 0 to 1 equal to 5.

Example 2:

Input: n = 3, edges = [[0,1,-1],[0,2,5]], source = 0, destination = 2, target = 6 Output: [] Explanation: The graph above contains the initial edges. It is not possible to make the distance from 0 to 2 equal to 6 by modifying the edge with weight -1. So, an empty array is returned.

Example 3:

Input: n = 4, edges = [[1,0,4],[1,2,3],[2,3,5],[0,3,-1]], source = 0, destination = 2, target = 6 Output: [[1,0,4],[1,2,3],[2,3,5],[0,3,1]] Explanation: The graph above shows a modified graph having the shortest distance from 0 to 2 as 6.

Constraints:

1 <= n <= 100 1 <= edges.length <= n * (n - 1) / 2 edges[i].length == 3 0 <= ai, bi < n wi = -1 or 1 <= wi <= 107 ai != bi 0 <= source, destination < n source != destination 1 <= target <= 109 The graph is connected, and there are no self-loops or repeated edges

==
Problem Number: 2147 URL: https://leetcode.com/problems/greatest-

common-divisor-traversal Title: 2709. Greatest Common Divisor Traversal Problem Description: You are given a 0-indexed integer array nums, and you are allowed to traverse between its indices. You can traverse between index i and index j, i != j, if and only if gcd(nums[i], nums[j]) > 1, where gcd is the greatest common divisor. Your task is to determine if for every pair of indices i and j in nums, where i < j, there exists a sequence of traversals that can take us from i to j. Return true if it is possible to traverse between all such pairs of indices, or false otherwise.   Example 1: Input: nums = [2,3,6] Output: true Explanation: In this example, there are 3 possible pairs of indices: (0, 1), (0, 2), and (1, 2). To go from index 0 to index 1, we can use the sequence of traversals 0 -> 2 -> 1, where we move from index 0 to index 2 because gcd(nums[0], nums[2]) = gcd(2, 6) = 2 > 1, and then move from index 2 to index 1 because gcd(nums[2], nums[1]) = gcd(6, 3) = 3 > 1. To go from index 0 to index 2, we can just go directly because gcd(nums[0], nums[2]) = gcd(2, 6) = 2 > 1. Likewise, to go from index 1 to index 2, we can just go directly because gcd(nums[1], nums[2]) = gcd(3, 6) = 3 > 1.

Example 2: Input: nums = [3,9,5] Output: false Explanation: No sequence of traversals can take us from index 0 to index 2 in this example. So, we return false.

Example 3: Input: nums = [4,3,12,8] Output: true Explanation: There are 6 possible pairs of indices to traverse between: (0, 1), (0, 2), (0, 3), (1, 2), (1, 3), and (2, 3). A valid sequence of traversals exists for each pair, so we return true.

Constraints:

1 <= nums.length <= 105 1 <= nums[i] <= 105

==
Problem Number: 2148 URL: https://leetcode.com/problems/maximum-strictly-increasing-cells-in-a-matrix Title: 2713. Maximum Strictly Increasing Cells in a Matrix Problem Description: Given a 1-indexed m x n integer matrix mat, you can select any cell in the matrix as your starting cell. From the starting cell, you can move to any other cell in the same row or column, but only if the value of the destination cell is strictly greater than the value of the current cell. You can repeat this process as many times as possible, moving from cell to cell until you can no longer make any moves. Your task is to find the maximum number of cells that you can visit in the matrix by starting from some cell. Return an integer denoting the maximum number of cells that can be visited.   Example 1:

Input: mat = [[3,1],[3,4]] Output: 2 Explanation: The image shows how we can visit 2 cells starting from row 1, column 2. It can be shown that we cannot visit more than 2 cells no matter where we start from, so the answer is 2.

Example 2:

Input: mat = [[1,1],[1,1]] Output: 1 Explanation: Since the cells must be strictly increasing, we can only visit one cell in this example.

Example 3:

Input: mat = [[3,1,6],[-9,5,7]] Output: 4 Explanation: The image above shows how we can visit 4 cells starting from row 2, column 1. It can be shown that we cannot visit more than 4 cells no matter where we start from, so the answer is 4.

Constraints:

m == mat.length n == mat[i].length 1 <= m, n <= 105 1 <= m * n <= 105 -105 <= mat[i][j] <= 105

==
Problem Number: 2149 URL: https://leetcode.com/problems/count-of-integers Title: 2719. Count of Integers Problem Description: You are given two numeric strings num1 and num2 and two integers max_sum and min_sum. We denote an integer x to be good if:

num1 <= x <= num2 min_sum <= digit_sum(x) <= max_sum.

Return the number of good integers. Since the answer may be large, return it modulo 109 + 7. Note that digit_sum(x) denotes the sum of the digits of x. Example 1: Input: num1 = "1", num2 = "12", min_sum = 1, max_sum = 8 Output: 11 Explanation: There are 11 integers whose sum of digits lies between 1 and 8 are 1,2,3,4,5,6,7,8,10,11, and 12. Thus, we return 11.

Example 2: Input: num1 = "1", num2 = "5", min_sum = 1, max_sum = 5 Output: 5 Explanation: The 5 integers whose sum of digits lies between 1 and 5 are 1,2,3,4, and 5. Thus, we return 5.

Constraints:

1 <= num1 <= num2 <= 1022 1 <= min_sum <= max_sum <= 400

==
Problem Number: 2150 URL: https://leetcode.com/problems/find-a-good-subset-of-the-matrix Title: 2732. Find a Good Subset of the Matrix Problem Description: You are given a 0-indexed m x n binary matrix grid. Let us call a non-empty subset of rows good if the sum of each column of the subset is at most half of the length of the subset. More formally, if the length of the chosen subset of rows is k, then the sum of each column should be at most floor(k / 2). Return an integer array that contains row indices of a good subset sorted in ascending order. If there are multiple good subsets, you can return any of them. If there are no good subsets, return an empty array. A subset of rows of the matrix grid is any matrix that can be obtained by deleting some (possibly none or all) rows from grid.   Example 1: Input: grid = [[0,1,1,0],[0,0,0,1],[1,1,1,1]] Output: [0,1] Explanation: We can choose the 0th and 1st rows to create a good subset of rows. The length of the chosen subset is 2. - The sum of the 0th column is 0 + 0 = 0, which is at most half of the length of the subset. - The sum of the 1st column is 1 + 0 = 1, which is at most half of the length of the subset. - The sum of the 2nd column is 1 + 0 = 1, which is at most half of

the length of the subset. - The sum of the 3rd column is 0 + 1 = 1, which is at most half of the length of the subset.

Example 2: Input: grid = [[0]] Output: [0] Explanation: We can choose the 0th row to create a good subset of rows. The length of the chosen subset is 1. - The sum of the 0th column is 0, which is at most half of the length of the subset.

Example 3: Input: grid = [[1,1,1],[1,1,1]] Output: [] Explanation: It is impossible to choose any subset of rows to create a good subset.

Constraints:

m == grid.length n == grid[i].length 1 <= m <= 104 1 <= n <= 5 grid[i][j] is either 0 or 1.

==
Problem Number: 2151 URL: https://leetcode.com/problems/maximum-sum-queries Title: 2736. Maximum Sum Queries Problem Description: You are given two 0-indexed integer arrays nums1 and nums2, each of length n, and a 1-indexed 2D array queries where queries[i] = [xi, yi]. For the ith query, find the maximum value of nums1[j] + nums2[j] among all indices j (0 <= j < n), where nums1[j] >= xi and nums2[j] >= yi, or -1 if there is no j satisfying the constraints. Return an array answer where answer[i] is the answer to the ith query. Example 1: Input: nums1 = [4,3,1,2], nums2 = [2,4,9,5], queries = [[4,1],[1,3],[2,5]] Output: [6,10,7] Explanation: For the 1st query xi = 4 and yi = 1, we can select index j = 0 since nums1[j] >= 4 and nums2[j] >= 1. The sum nums1[j] + nums2[j] is 6, and we can show that 6 is the maximum we can obtain.

For the 2nd query xi = 1 and yi = 3, we can select index j = 2 since nums1[j] >= 1 and nums2[j] >= 3. The sum nums1[j] + nums2[j] is 10, and we can show that 10 is the maximum we can obtain.

For the 3rd query xi = 2 and yi = 5, we can select index j = 3 since nums1[j] >= 2 and nums2[j] >= 5. The sum nums1[j] + nums2[j] is 7, and we can show that 7 is the maximum we can obtain.

Therefore, we return [6,10,7].

Example 2: Input: nums1 = [3,2,5], nums2 = [2,3,4], queries = [[4,4],[3,2],[1,1]] Output: [9,9,9] Explanation: For this example, we can use index j = 2 for all the queries since it satisfies the constraints for each query.

Example 3: Input: nums1 = [2,1], nums2 = [2,3], queries = [[3,3]] Output: [-1] Explanation: There is one query in this example with xi = 3 and yi = 3. For every index, j, either nums1[j] < xi or nums2[j] < yi. Hence, there is no solution.

Constraints:

nums1.length == nums2.length n == nums1.length 1 <= n <= 105 1 <= nums1[i], nums2[i] <= 109 1 <= queries.length <= 105 queries[i].length == 2 xi == queries[i][1] yi == queries[i][2] 1 <= xi, yi <= 109

==

Problem Number: 2152 URL: https://leetcode.com/problems/painting-the-walls Title: 2742. Painting the Walls Problem Description: You are given two 0-indexed integer arrays, cost and time, of size n representing the costs and the time taken to paint n different walls respectively. There are two painters available:

A paid painter that paints the ith wall in time[i] units of time and takes cost[i] units of money. A free painter that paints any wall in 1 unit of time at a cost of 0. But the free painter can only be used if the paid painter is already occupied.

Return the minimum amount of money required to paint the n walls. Example 1: Input: cost = [1,2,3,2], time = [1,2,3,2] Output: 3 Explanation: The walls at index 0 and 1 will be painted by the paid painter, and it will take 3 units of time; meanwhile, the free painter will paint the walls at index 2 and 3, free of cost in 2 units of time. Thus, the total cost is 1 + 2 = 3.

Example 2: Input: cost = [2,3,4,2], time = [1,1,1,1] Output: 4 Explanation: The walls at index 0 and 3 will be painted by the paid painter, and it will take 2 units of time; meanwhile, the free painter will paint the walls at index 1 and 2, free of cost in 2 units of time. Thus, the total cost is 2 + 2 = 4.

Constraints:

1 <= cost.length <= 500 cost.length == time.length 1 <= cost[i] <= 106 1 <= time[i] <= 500

==

Problem Number: 2153 URL: https://leetcode.com/problems/robot-collisions Title: 2751. Robot Collisions Problem Description: There are n 1-indexed robots, each having a position on a line, health, and movement direction. You are given 0-indexed integer arrays positions, healths, and a string directions (directions[i] is either 'L' for left or 'R' for right). All integers in positions are unique. All robots start moving on the line simultaneously at the same speed in their given directions. If two robots ever share the same position while moving, they will collide. If two robots collide, the robot with lower health is removed from the line, and the health of the other robot decreases by one. The surviving robot continues in the same direction it was going. If both robots have the same health, they are both removed from the line. Your task is to determine the health of the robots that survive the collisions, in the same order that the robots were given, i.e. final heath of robot 1 (if survived), final health of robot 2 (if survived), and so on. If there are no survivors, return an empty array. Return an array containing the health of the remaining robots (in the order they were given in the input), after no further collisions can occur. Note: The positions may be unsorted. Example 1:

Input: positions = [5,4,3,2,1], healths = [2,17,9,15,10], directions = "RRRRR" Output: [2,17,9,15,10] Explanation: No collision occurs in this example, since all robots are moving in the same direction. So, the health of the robots in order from the first robot is returned, [2, 17, 9, 15, 10].

Example 2:

Input: positions = [3,5,2,6], healths = [10,10,15,12], directions = ”RLRL” Output: [14] Explanation: There are 2 collisions in this example. Firstly, robot 1 and robot 2 will collide, and since both have the same health, they will be removed from the line. Next, robot 3 and robot 4 will collide and since robot 4’s health is smaller, it gets removed, and robot 3’s health becomes 15 - 1 = 14. Only robot 3 remains, so we return [14].

Example 3:

Input: positions = [1,2,5,6], healths = [10,10,11,11], directions = ”RLRL” Output: [] Explanation: Robot 1 and robot 2 will collide and since both have the same health, they are both removed. Robot 3 and 4 will collide and since both have the same health, they are both removed. So, we return an empty array, []. Constraints:

1 <= positions.length == healths.length == directions.length == n <= 105 1 <= positions[i], healths[i] <= 109 directions[i] == ’L’ or directions[i] == ’R’ All values in positions are distinct

==
Problem Number: 2154 URL: https://leetcode.com/problems/sum-of-imbalance-numbers-of-all-subarrays Title: 2763. Sum of Imbalance Numbers of All Subarrays Problem Description: The imbalance number of a 0-indexed integer array arr of length n is defined as the number of indices in sarr = sorted(arr) such that:

0 <= i < n - 1, and sarr[i+1] - sarr[i] > 1

Here, sorted(arr) is the function that returns the sorted version of arr. Given a 0-indexed integer array nums, return the sum of imbalance numbers of all its subarrays. A subarray is a contiguous non-empty sequence of elements within an array. Example 1: Input: nums = [2,3,1,4] Output: 3 Explanation: There are 3 subarrays with non-zero imbalance numbers: - Subarray [3, 1] with an imbalance number of 1. - Subarray [3, 1, 4] with an imbalance number of 1. - Subarray [1, 4] with an imbalance number of 1. The imbalance number of all other subarrays is 0. Hence, the sum of imbalance numbers of all the subarrays of nums is 3.

Example 2: Input: nums = [1,3,3,3,5] Output: 8 Explanation: There are 7 subarrays with non-zero imbalance numbers: - Subarray [1, 3] with an imbalance number of 1. - Subarray [1, 3, 3] with an imbalance number of 1. - Subarray [1, 3, 3, 3] with an imbalance number of 1. - Subarray [1, 3, 3, 3, 5] with an imbalance number of 2. - Subarray [3, 3, 3, 5] with an imbalance number of 1. - Subarray [3, 3, 5] with an imbalance number of 1. - Subarray [3, 5] with an imbalance number of 1. The imbalance number of all other subarrays is 0. Hence, the sum of imbalance numbers of all the subarrays of nums is 8. Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= nums.length

==
Problem Number: 2155 URL: https://leetcode.com/problems/length-of-the-longest-valid-substring Title: 2781. Length of the Longest Valid Substring Problem Description: You are given a string word and an array of strings forbidden. A string is called valid if none of its substrings are present in forbidden. Return the length of the longest valid substring of the string word. A substring is a contiguous sequence of characters in a string, possibly empty. Example 1: Input: word = "cbaaaabc", forbidden = ["aaa","cb"] Output: 4 Explanation: There are 11 valid substrings in word: "c", "b", "a", "ba", "aa", "bc", "baa", "aab", "ab", "abc" and "aabc". The length of the longest valid substring is 4. It can be shown that all other substrings contain either "aaa" or "cb" as a substring. Example 2: Input: word = "leetcode", forbidden = ["de","le","e"] Output: 4 Explanation: There are 11 valid substrings in word: "l", "t", "c", "o", "d", "tc", "co", "od", "tco", "cod", and "tcod". The length of the longest valid substring is 4. It can be shown that all other substrings contain either "de", "le", or "e" as a substring.

Constraints:

1 <= word.length <= 105 word consists only of lowercase English letters. 1 <= forbidden.length <= 105 1 <= forbidden[i].length <= 10 forbidden[i] consists only of lowercase English letters.

==
Problem Number: 2156 URL: https://leetcode.com/problems/maximum-number-of-groups-with-increasing-length Title: 2790. Maximum Number of Groups With Increasing Length Problem Description: You are given a 0-indexed array usageLimits of length n. Your task is to create groups using numbers from 0 to n - 1, ensuring that each number, i, is used no more than usageLimits[i] times in total across all groups. You must also satisfy the following conditions:

Each group must consist of distinct numbers, meaning that no duplicate numbers are allowed within a single group. Each group (except the first one) must have a length strictly greater than the previous group.

Return an integer denoting the maximum number of groups you can create while satisfying these conditions. Example 1: Input: usageLimits = [1,2,5] Output: 3 Explanation: In this example, we can use 0 at most once, 1 at most twice, and 2 at most five times. One way of creating the maximum number of groups while satisfying the conditions is: Group 1 contains the number [2]. Group 2 contains the numbers [1,2]. Group 3 contains the numbers [0,1,2]. It can be shown that the maximum number of groups is 3. So, the output is 3. Example 2: Input: usageLimits = [2,1,2] Output: 2 Explanation: In this example, we can use 0 at most twice, 1 at most once, and 2 at most twice. One way of creating the maximum number of groups while satisfying the conditions is: Group 1 contains the number [0]. Group 2 contains the numbers [1,2]. It can be shown that the maximum number of groups is 2. So, the output is 2.

Example 3: Input: usageLimits = [1,1] Output: 1 Explanation: In this example, we can use both 0 and 1 at most once. One way of creating the maximum number of groups while satisfying the conditions is: Group 1 contains the number [0]. It can be shown that the maximum number of groups is 1. So, the output is 1.

Constraints:

1 <= usageLimits.length <= 105 1 <= usageLimits[i] <= 109

==

Problem Number: 2157 URL: https://leetcode.com/problems/count-paths-that-can-form-a-palindrome-in-a-tree Title: 2791. Count Paths That Can Form a Palindrome in a Tree Problem Description: You are given a tree (i.e. a connected, undirected graph that has no cycles) rooted at node 0 consisting of n nodes numbered from 0 to n - 1. The tree is represented by a 0-indexed array parent of size n, where parent[i] is the parent of node i. Since node 0 is the root, parent[0] == -1. You are also given a string s of length n, where s[i] is the character assigned to the edge between i and parent[i]. s[0] can be ignored. Return the number of pairs of nodes (u, v) such that u < v and the characters assigned to edges on the path from u to v can be rearranged to form a palindrome. A string is a palindrome when it reads the same backwards as forwards. Example 1:

Input: parent = [-1,0,0,1,1,2], s = "acaabc" Output: 8 Explanation: The valid pairs are: - All the pairs (0,1), (0,2), (1,3), (1,4) and (2,5) result in one character which is always a palindrome. - The pair (2,3) result in the string "aca" which is a palindrome. - The pair (1,5) result in the string "cac" which is a palindrome. - The pair (3,5) result in the string "acac" which can be rearranged into the palindrome "acca".

Example 2: Input: parent = [-1,0,0,0,0], s = "aaaaa" Output: 10 Explanation: Any pair of nodes (u,v) where u < v is valid.

Constraints:

n == parent.length == s.length 1 <= n <= 105 0 <= parent[i] <= n - 1 for all i >= 1 parent[0] == -1 parent represents a valid tree. s consists of only lowercase English letters.

==

Problem Number: 2158 URL: https://leetcode.com/problems/count-stepping-numbers-in-range Title: 2801. Count Stepping Numbers in Range Problem Description: Given two positive integers low and high represented as strings, find the count of stepping numbers in the inclusive range [low, high]. A stepping number is an integer such that all of its adjacent digits have an absolute difference of exactly 1. Return an integer denoting the count of stepping numbers in the inclusive range [low, high]. Since the answer may be very large, return it modulo 109 + 7. Note: A stepping number should not have a leading zero. Example 1: Input: low = ”1”, high = ”11” Output: 10 Explanation: The stepping numbers in the range [1,11] are 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. There are a total of 10 stepping numbers in the range. Hence, the output is 10. Example 2: Input: low = ”90”, high = ”101” Output: 2 Explanation: The stepping numbers in the range [90,101] are 98 and 101. There are a total of 2 stepping numbers in the range. Hence, the output is 2. Constraints:

1 <= int(low) <= int(high) < 10100 1 <= low.length, high.length <= 100 low and high consist of only digits. low and high don't have any leading zeros.

==
Problem Number: 2159 URL: https://leetcode.com/problems/minimum-time-to-make-array-sum-at-most-x Title: 2809. Minimum Time to Make Array Sum At Most x Problem Description: You are given two 0-indexed integer arrays nums1 and nums2 of equal length. Every second, for all indices 0 <= i < nums1.length, value of nums1[i] is incremented by nums2[i]. After this is done, you can do the following operation:

Choose an index 0 <= i < nums1.length and make nums1[i] = 0.

You are also given an integer x. Return the minimum time in which you can make the sum of all elements of nums1 to be less than or equal to x, or -1 if this is not possible. Example 1: Input: nums1 = [1,2,3], nums2 = [1,2,3], x = 4 Output: 3 Explanation: For the 1st second, we apply the operation on i = 0. Therefore nums1 = [0,2+2,3+3] = [0,4,6]. For the 2nd second, we apply the operation on i = 1. Therefore nums1 = [0+1,0,6+3] = [1,0,9]. For the 3rd second, we apply the operation on i = 2. Therefore nums1 = [1+1,0+2,0] = [2,2,0]. Now sum of nums1 = 4. It can be shown that these operations are optimal, so we return 3.

Example 2: Input: nums1 = [1,2,3], nums2 = [3,3,3], x = 4 Output: -1 Explanation: It can be shown that the sum of nums1 will always be greater than x, no matter which operations are performed.

Constraints:

1 <= nums1.length <= 103 1 <= nums1[i] <= 103 0 <= nums2[i] <= 103 nums1.length == nums2.length 0 <= x <= 106

==
Problem Number: 2160 URL: https://leetcode.com/problems/maximum-elegance-of-a-k-length-subsequence Title: 2813. Maximum Elegance of a K-Length Subsequence Problem Description: You are given a 0-indexed 2D integer array items of length n and an integer k. items[i] = [profiti, categoryi], where profiti and categoryi denote the profit and category of the ith item respectively. Let's define the elegance of a subsequence of items as total_profit

+ distinct_categories2, where total_profit is the sum of all profits in the subsequence, and distinct_categories is the number of distinct categories from all the categories in the selected subsequence. Your task is to find the maximum elegance from all subsequences of size k in items. Return an integer denoting the maximum elegance of a subsequence of items with size exactly k. Note: A subsequence of an array is a new array generated from the original array by deleting some elements (possibly none) without changing the remaining elements' relative order. Example 1: Input: items = [[3,2],[5,1],[10,1]], k = 2 Output: 17 Explanation: In this example, we have to select a subsequence of size 2. We can select items[0] = [3,2] and items[2] = [10,1]. The total profit in this subsequence is 3 + 10 = 13, and the subsequence contains 2 distinct categories [2,1]. Hence, the elegance is 13 + 22 = 17, and we can show that it is the maximum achievable elegance.

Example 2: Input: items = [[3,1],[3,1],[2,2],[5,3]], k = 3 Output: 19 Explanation: In this example, we have to select a subsequence of size 3. We can select items[0] = [3,1], items[2] = [2,2], and items[3] = [5,3]. The total profit in this subsequence is 3 + 2 + 5 = 10, and the subsequence contains 3 distinct categories [1,2,3]. Hence, the elegance is 10 + 32 = 19, and we can show that it is the maximum achievable elegance. Example 3: Input: items = [[1,1],[2,1],[3,1]], k = 3 Output: 7 Explanation: In this example, we have to select a subsequence of size 3. We should select all the items. The total profit will be 1 + 2 + 3 = 6, and the subsequence contains 1 distinct category [1]. Hence, the maximum elegance is 6 + 12 = 7. Constraints:

1 <= items.length == n <= 105 items[i].length == 2 items[i][0] == profiti items[i][1] == categoryi 1 <= profiti <= 109 1 <= categoryi <= n 1 <= k <= n

==
Problem Number: 2161 URL: https://leetcode.com/problems/apply-operations-to-maximize-score Title: 2818. Apply Operations to Maximize Score Problem Description: You are given an array nums of n positive integers and an integer k. Initially, you start with a score of 1. You have to maximize your score by applying the following operation at most k times:

Choose any non-empty subarray nums[l, ..., r] that you haven't chosen previously. Choose an element x of nums[l, ..., r] with the highest prime score. If multiple such elements exist, choose the one with the smallest index. Multiply your score by x.

Here, nums[l, ..., r] denotes the subarray of nums starting at index l and ending at the index r, both ends being inclusive. The prime score of an integer x is equal to the number of distinct prime factors of x. For example, the prime score of 300 is 3 since 300 = 2 * 2 * 3 * 5 * 5. Return the maximum possible score after applying at most k operations. Since the answer may be large, return it modulo 109 + 7. Example 1: Input: nums = [8,3,9,3,8], k = 2 Output: 81 Explanation: To get a score of 81, we can apply the following operations: - Choose subarray nums[2, ..., 2]. nums[2] is the only element in this subarray. Hence, we multiply the score by nums[2]. The score becomes 1 * 9 = 9. - Choose subarray nums[2, ..., 3]. Both nums[2] and nums[3] have a prime score of 1, but nums[2] has the smaller index. Hence, we multiply the score by nums[2]. The score becomes 9 * 9 = 81. It can be proven that 81 is the highest score one can obtain. Example 2: Input: nums = [19,12,14,6,10,18], k = 3 Output: 4788 Explanation: To get a score of 4788, we can apply the following operations: - Choose subarray nums[0, ..., 0]. nums[0] is the only element in this subarray. Hence, we multiply the score by nums[0]. The score becomes 1 * 19 = 19. - Choose subarray nums[5, ..., 5]. nums[5] is the only element in this subarray. Hence, we multiply the score by nums[5]. The score becomes 19 * 18 = 342. - Choose subarray nums[2, ..., 3]. Both nums[2] and nums[3] have a prime score of 2, but nums[2] has the smaller index. Hence, we multipy the score by nums[2]. The score becomes 342 * 14 = 4788. It can be proven that 4788 is the highest score one can obtain.

Constraints:

1 <= nums.length == n <= 105 1 <= nums[i] <= 105 1 <= k <= min(n * (n + 1) / 2, 109)

==

Problem Number: 2162 URL: https://leetcode.com/problems/number-of-beautiful-integers-in-the-range Title: 2827. Number of Beautiful Integers in the Range Problem Description: You are given positive integers low, high, and k. A number is beautiful if it meets both of the following conditions:

The count of even digits in the number is equal to the count of odd digits. The number is divisible by k.

Return the number of beautiful integers in the range [low, high]. Example 1: Input: low = 10, high = 20, k = 3 Output: 2 Explanation: There are 2 beautiful integers in the given range: [12,18]. - 12 is beautiful because it contains 1 odd digit and 1 even digit, and is divisible by k = 3. - 18 is beautiful because it contains 1 odd digit and 1 even digit, and is divisible by k = 3. Additionally we can see that: - 16 is not beautiful because it is not divisible by k = 3. - 15 is not beautiful because it does not contain equal counts even and odd digits. It can be shown that there are only 2 beautiful integers in the given range.

Example 2: Input: low = 1, high = 10, k = 1 Output: 1 Explanation: There is 1 beautiful integer in the given range: [10]. - 10 is beautiful because it contains 1 odd digit and 1 even digit, and is divisible by k = 1. It can be shown that there is only 1 beautiful integer in the given range.

Example 3: Input: low = 5, high = 5, k = 2 Output: 0 Explanation: There are 0 beautiful integers in the given range. - 5 is not beautiful because it is not divisible by k = 2 and it does not contain equal even and odd digits.

Constraints:

0 < low <= high <= 109 0 < k <= 20

==
Problem Number: 2163 URL: https://leetcode.com/problems/minimum-operations-to-form-subsequence-with-target-sum Title: 2835. Minimum Operations to Form Subsequence With Target Sum Problem Description: You are given a 0-indexed array nums consisting of non-negative powers of 2, and an integer target. In one operation, you must apply the following changes to the array:

Choose any element of the array nums[i] such that nums[i] > 1. Remove nums[i] from the array. Add two occurrences of nums[i] / 2 to the end of nums.

Return the minimum number of operations you need to perform so that nums contains a subsequence whose elements sum to target. If it is impossible to obtain such a subsequence, return -1. A subsequence is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.   Example 1: Input: nums = [1,2,8], target = 7 Output: 1 Explanation: In the first operation, we choose element nums[2]. The array becomes equal to nums = [1,2,4,4]. At this stage, nums contains the subsequence [1,2,4] which sums up to 7. It can be shown that there is no shorter sequence of operations that results in a subsequnce that sums up to 7.

Example 2: Input: nums = [1,32,1,2], target = 12 Output: 2 Explanation: In the first operation, we choose element nums[1]. The array becomes equal to nums = [1,1,2,16,16]. In the second operation, we choose element nums[3]. The array becomes equal to nums = [1,1,2,16,8,8] At this stage, nums contains the subsequence [1,1,2,8] which sums up to 12. It can be shown that there is no shorter sequence of operations that results in a subsequence that sums up to 12. Example 3: Input: nums = [1,32,1], target = 35 Output: -1 Explanation: It can be shown that no sequence of operations results in a subsequence that sums up to 35.

Constraints:

1 <= nums.length <= 1000 1 <= nums[i] <= 230 nums consists only of non-negative powers of two. 1 <= target < 231

==
Problem Number: 2164 URL: https://leetcode.com/problems/maximize-value-of-function-in-a-ball-passing-game Title: 2836. Maximize Value of Function in a Ball Passing Game Problem Description: You are given a 0-indexed integer array receiver of length n and an integer k. There are n players having a unique id in the range [0, n - 1] who will play a ball passing game, and receiver[i] is the id of the player who receives passes from the player with id i. Players can pass to themselves, i.e. receiver[i] may be equal to i. You must choose one of the n players as the starting player for the game, and the ball will be passed exactly k times starting from the chosen player. For a chosen starting player having id x, we define a function f(x) that denotes the sum of x and the ids of all players who receive the ball during the k passes, including repetitions. In other words, f(x) = x + receiver[x] + receiver[receiver[x]] + ... + receiver(k)[x]. Your

task is to choose a starting player having id x that maximizes the value of f(x). Return an integer denoting the maximum value of the function. Note: receiver may contain duplicates.   Example 1: Pass NumberSender IDReceiver IDx + Receiver IDs   21213210330254216 Input: receiver = [2,0,1], k = 4 Output: 6 Explanation: The table above shows a simulation of the game starting with the player having id x = 2. From the table, f(2) is equal to 6. It can be shown that 6 is the maximum achievable value of the function. Hence, the output is 6.

Example 2: Pass NumberSender IDReceiver IDx + Receiver IDs 41437232932110 Input: receiver = [1,1,1,2,3], k = 3 Output: 10 Explanation: The table above shows a simulation of the game starting with the player having id x = 4. From the table, f(4) is equal to 10. It can be shown that 10 is the maximum achievable value of the function. Hence, the output is 10.

Constraints:

1 <= receiver.length == n <= 105 0 <= receiver[i] <= n - 1 1 <= k <= 1010

==

Problem Number: 2165 URL: https://leetcode.com/problems/count-k-subsequences-of-a-string-with-maximum-beauty Title: 2842. Count K-Subsequences of a String With Maximum Beauty Problem Description: You are given a string s and an integer k. A k-subsequence is a subsequence of s, having length k, and all its characters are unique, i.e., every character occurs once. Let f(c) denote the number of times the character c occurs in s. The beauty of a k-subsequence is the sum of f(c) for every character c in the k-subsequence. For example, consider s = "abbbdd" and k = 2:

f('a') = 1, f('b') = 3, f('d') = 2 Some k-subsequences of s are:

"abbbdd" -> "ab" having a beauty of f('a') + f('b') = 4 "abbbdd" -> "ad" having a beauty of f('a') + f('d') = 3 "abbbdd" -> "bd" having a beauty of f('b') + f('d') = 5

Return an integer denoting the number of k-subsequences whose beauty is the maximum among all k-subsequences. Since the answer may be too large, return it modulo 109 + 7. A subsequence of a string is a new string formed from the original string by deleting some (possibly none) of the characters without disturbing the relative positions of the remaining characters. Notes

f(c) is the number of times a character c occurs in s, not a k-subsequence. Two k-subsequences are considered different if one is formed by an index that is not present in the other. So, two k-subsequences may form the same string.

Example 1: Input: s = "bcca", k = 2 Output: 4 Explanation: From s we have f('a') = 1, f('b') = 1, and f('c') = 2. The k-subsequences of s are: bcca having a beauty of f('b') + f('c') = 3 bcca having a beauty of f('b') + f('c') = 3 bcca having a beauty of f('b') + f('a') = 2 bcca having a beauty of f('c') + f('a') = 3 bcca having a beauty of f('c') + f('a') = 3 There are 4 k-subsequences that have the maximum beauty, 3. Hence, the answer is 4.

Example 2: Input: s = "abbcd", k = 4 Output: 2 Explanation: From s we have f('a') = 1, f('b') = 2, f('c') = 1, and f('d') = 1. The k-subsequences of s are: abbcd having a beauty of f('a') + f('b') + f('c') + f('d') = 5 abbcd having a beauty of f('a') + f('b') + f('c') + f('d') = 5 There are 2 k-subsequences that have the maximum beauty, 5. Hence, the answer is 2.

Constraints:

1 <= s.length <= 2 * 105 1 <= k <= s.length s consists only of lowercase English letters.

==
Problem Number: 2166 URL: https://leetcode.com/problems/minimum-edge-weight-equilibrium-queries-in-a-tree Title: 2846. Minimum Edge Weight Equilibrium Queries in a Tree Problem Description: There is an undirected tree with n nodes labeled from 0 to n - 1. You are given the integer n and a 2D integer array edges of length n - 1, where edges[i] = [ui, vi, wi] indicates that there is an edge between nodes ui and vi with weight wi in the tree. You are also given a 2D integer array queries of length m, where queries[i] = [ai, bi]. For each query, find the minimum number of operations required to make the weight of every edge on the path from ai to bi equal. In one operation, you can choose any edge of the tree and change its weight to any value. Note that:

Queries are independent of each other, meaning that the tree returns to its initial state on each new query. The path from ai to bi is a sequence of distinct nodes starting with node ai and ending with node bi such that every two adjacent nodes in the sequence share an edge in the tree.

Return an array answer of length m where answer[i] is the answer to the ith query. Example 1:

Input: n = 7, edges = [[0,1,1],[1,2,1],[2,3,1],[3,4,2],[4,5,2],[5,6,2]], queries = [[0,3],[3,6],[2,6],[0,6]] Output: [0,0,1,3] Explanation: In the first query, all the edges in the path from 0 to 3 have a weight of 1. Hence, the answer is 0. In the second query, all the edges in the path from 3 to 6 have a weight of 2. Hence, the answer is 0. In the third query, we change the weight of edge [2,3] to 2. After this operation, all the edges in the path from 2 to 6 have a weight of 2. Hence, the answer is 1. In the fourth query, we change the weights of edges [0,1], [1,2] and [2,3] to 2. After these operations, all the edges in the path from 0 to 6 have a weight of 2. Hence, the answer is 3. For each queries[i], it can be shown that answer[i] is the minimum number of operations needed to equalize all the edge weights in the path from ai to bi.

Example 2:

Input: n = 8, edges = [[1,2,6],[1,3,4],[2,4,6],[2,5,3],[3,6,6],[3,0,8],[7,0,2]], queries = [[4,6],[0,4],[6,5],[7,4]] Output: [1,2,2,3] Explanation: In the first query, we change the weight of edge [1,3] to 6. After this operation, all the edges in the path from 4 to 6 have a weight of 6. Hence, the answer is 1. In the second query, we change the weight of edges [0,3] and [3,1] to 6. After these operations, all the edges in the path from 0 to 4 have a weight of 6. Hence, the answer is 2. In the third query, we change the weight of edges [1,3] and [5,2] to 6. After these operations, all the edges in the path from 6 to 5 have a weight of 6. Hence, the answer is 2. In the fourth query, we change the weights of edges [0,7], [0,3] and [1,3] to 6. After these operations, all the edges in the path from 7 to 4 have a weight of 6. Hence, the answer is 3. For each queries[i], it can be shown that answer[i] is the minimum number of operations needed to equalize all the edge weights in the path from ai to bi.

Constraints:

1 <= n <= 104 edges.length == n - 1 edges[i].length == 3 0 <= ui, vi < n 1 <= wi <= 26 The input is generated such that edges represents a valid tree. 1 <= queries.length == m <= 2 * 104 queries[i].length == 2 0 <= ai, bi < n

==
Problem Number: 2167 URL: https://leetcode.com/problems/string-transformation Title: 2851. String Transformation Problem Description: You are given two strings s and t of equal length n. You can perform the following operation on the string s:

Remove a suffix of s of length l where 0 < l < n and append it at the start of s. For example, let s = 'abcd' then in one operation you can remove the suffix 'cd' and append it in front of s making s = 'cdab'.

You are also given an integer k. Return the number of ways in which s can be transformed into t in exactly k operations. Since the answer can be large, return it modulo 109 + 7. Example 1: Input: s = "abcd", t = "cdab", k = 2 Output: 2 Explanation: First way: In first operation, choose suffix from index = 3, so resulting s = "dabc". In second operation, choose suffix from index = 3, so resulting s = "cdab".

Second way: In first operation, choose suffix from index = 1, so resulting s = "bcda". In second operation, choose suffix from index = 1, so resulting s = "cdab".

Example 2: Input: s = "ababab", t = "ababab", k = 1 Output: 2 Explanation: First way: Choose suffix from index = 2, so resulting s = "ababab".

Second way: Choose suffix from index = 4, so resulting s = "ababab".

Constraints:

2 <= s.length <= 5 * 105 1 <= k <= 1015 s.length == t.length s and t consist of only lowercase English alphabets.

==
Problem Number: 2168 URL: https://leetcode.com/problems/minimum-edge-reversals-so-every-node-is-reachable Title: 2858. Minimum Edge Reversals So Every Node Is Reachable Problem Description: There is a simple directed graph with n nodes labeled from 0 to n - 1. The graph would form a tree if its

edges were bi-directional. You are given an integer n and a 2D integer array edges, where edges[i] = [ui, vi] represents a directed edge going from node ui to node vi. An edge reversal changes the direction of an edge, i.e., a directed edge going from node ui to node vi becomes a directed edge going from node vi to node ui. For every node i in the range [0, n - 1], your task is to independently calculate the minimum number of edge reversals required so it is possible to reach any other node starting from node i through a sequence of directed edges. Return an integer array answer, where answer[i] is the minimum number of edge reversals required so it is possible to reach any other node starting from node i through a sequence of directed edges.   Example 1:

Input: n = 4, edges = [[2,0],[2,1],[1,3]] Output: [1,1,0,2] Explanation: The image above shows the graph formed by the edges. For node 0: after reversing the edge [2,0], it is possible to reach any other node starting from node 0. So, answer[0] = 1. For node 1: after reversing the edge [2,1], it is possible to reach any other node starting from node 1. So, answer[1] = 1. For node 2: it is already possible to reach any other node starting from node 2. So, answer[2] = 0. For node 3: after reversing the edges [1,3] and [2,1], it is possible to reach any other node starting from node 3. So, answer[3] = 2.

Example 2:

Input: n = 3, edges = [[1,2],[2,0]] Output: [2,0,1] Explanation: The image above shows the graph formed by the edges. For node 0: after reversing the edges [2,0] and [1,2], it is possible to reach any other node starting from node 0. So, answer[0] = 2. For node 1: it is already possible to reach any other node starting from node 1. So, answer[1] = 0. For node 2: after reversing the edge [1, 2], it is possible to reach any other node starting from node 2. So, answer[2] = 1.

Constraints:

2 <= n <= 105 edges.length == n - 1 edges[i].length == 2 0 <= ui == edges[i][0] < n 0 <= vi == edges[i][1] < n ui != vi The input is generated such that if the edges were bi-directional, the graph would be a tree.

== Problem Number: 2169 URL: https://leetcode.com/problems/maximum-element-sum-of-a-complete-subset-of-indices Title: 2862. Maximum Element-Sum of a Complete Subset of Indices Problem Description: You are given a 1-indexed array nums of n integers. A set of numbers is complete if the product of every pair of its elements is a perfect square. For a subset of the indices set {1, 2, ..., n} represented as {i1, i2, ..., ik}, we define its element-sum as: nums[i1] + nums[i2] + ... + nums[ik]. Return the maximum element-sum of a complete subset of the indices set {1, 2, ..., n}. A perfect square is a number that can be expressed as the product of an integer by itself.   Example 1: Input: nums = [8,7,3,5,7,2,4,9] Output: 16 Explanation: Apart from the subsets consisting of a single index, there are two other complete subsets of indices: {1,4} and {2,8}. The sum of the elements corresponding to indices 1 and 4 is equal to nums[1] + nums[4] = 8 + 5 = 13. The sum of the elements corresponding to indices 2 and 8 is equal to nums[2] + nums[8] = 7 + 9 = 16. Hence, the maximum element-sum of a complete subset of indices is 16.

Example 2: Input: nums = [5,10,3,10,1,13,7,9,4] Output: 19 Explanation: Apart from the subsets consisting of a single index, there are four other complete subsets of indices: {1,4}, {1,9}, {2,8}, {4,9}, and {1,4,9}. The sum of the elements corresponding to indices 1 and 4 is equal to nums[1] + nums[4] = 5 + 10 = 15. The sum of the elements corresponding to indices 1 and 9 is equal to nums[1] + nums[9] = 5 + 4 = 9. The sum of the elements corresponding to indices 2 and 8 is equal to nums[2] + nums[8] = 10 + 9 = 19. The sum of the elements corresponding to indices 4 and 9 is equal to nums[4] + nums[9] = 10 + 4 = 14. The sum of the elements corresponding to indices 1, 4, and 9 is equal to nums[1] + nums[4] + nums[9] = 5 + 10 + 4 = 19. Hence, the maximum element-sum of a complete subset of indices is 19.

Constraints:

1 <= n == nums.length <= 104 1 <= nums[i] <= 109

==

Problem Number: 2170 URL: https://leetcode.com/problems/count-valid-paths-in-a-tree Title: 2867. Count Valid Paths in a Tree Problem Description: There is an undirected tree with n nodes labeled from 1 to n. You are given the integer n and a 2D integer array edges of length n - 1, where edges[i] = [ui, vi] indicates that there is an edge between nodes ui and vi in the tree. Return the number of valid paths in the tree. A path (a, b) is valid if there exists exactly one prime number among the node labels in the path from a to b. Note that:

The path (a, b) is a sequence of distinct nodes starting with node a and ending with node b such that every two adjacent nodes in the sequence share an edge in the tree. Path (a, b) and path (b, a) are considered the same and counted only once.

Example 1:

Input: n = 5, edges = [[1,2],[1,3],[2,4],[2,5]] Output: 4 Explanation: The pairs with exactly one prime number on the path between them are: - (1, 2) since the path from 1 to 2 contains prime number 2. - (1, 3) since the path from 1 to 3 contains prime number 3. - (1, 4) since the path from 1 to 4 contains prime number 2. - (2, 4) since the path from 2 to 4 contains prime number 2. It can be shown that there are only 4 valid paths.

Example 2:

Input: n = 6, edges = [[1,2],[1,3],[2,4],[3,5],[3,6]] Output: 6 Explanation: The pairs with exactly one prime number on the path between them are: - (1, 2) since the path from 1 to 2 contains prime number 2. - (1, 3) since the path from 1 to 3 contains prime number 3. - (1, 4) since the path from 1 to 4 contains prime number 2. - (1, 6) since the path from 1 to 6 contains prime number 3.

- (2, 4) since the path from 2 to 4 contains prime number 2. - (3, 6) since the path from 3 to 6 contains prime number 3. It can be shown that there are only 6 valid paths.

Constraints:

1 <= n <= 105 edges.length == n - 1 edges[i].length == 2 1 <= ui, vi <= n The input is generated such that edges represent a valid tree.

==
Problem Number: 1903 URL: https://www.lintcode.com/problem/3710 Title: Design Most Recently Used Queue Problem Description: DescriptionDesign a queue-like data structure that moves the most recently used elements to the end of the queue. Implement the MRUQueue class:

MRUQueue(int n): initialize the queue by [1,2,3,.... ,n] Construct MRUQueue. fetch(int k): return the kth element in the queue (indexed from 1) and move it to the end of the queue Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 k n 20001 \leq k \leq n \leq 20001 k n 2000 Call fetch() up to 2000 timesExampleExample 1: Input: MRUQueue(8) fetch(3) fetch(5) fetch(2) fetch(5) Output: [3, 6, 2, 8] Explanation: MRUQueue m = new MRUQueue(8); // initialize queue to [1,2,3,4,5,6,7,8] m.fetch(3); // return the 3rd element (3) and move it to the end of the queue [1,2,4,5,6,7,8,3] m.fetch(5); // return the 5th element (6) and move it to the end of the queue [1,2,4,5,7,8,3,6] m.fetch(2); // return the 2nd element (2) and move it to the end of the queue [1,4,5,7,8,3,6,2] m.fetch(5); // return the 5th element (8) and move it to the end of the queue [1,4,5,7,3,6,2,8] Example 2: Input: MRUQueue(8) fetch(1) fetch(1) fetch(1) fetch(1) Output: [1,2,3,4]
==
Problem Number: 1719 URL: https://www.lintcode.com/problem/3692 Title: Put Boxes Into the Warehouse II Problem Description: DescriptionGiven two arrays of positive integers boxes and warehouse, representing the heights of the boxes to be placed in the warehouse and the heights of the rooms in the warehouse, respectively. The rooms in the warehouse are numbered from 0 to n - 1 from left to right, and warehouse[i] denotes the height of the ith room. Now the boxes need to be pushed into the warehouse, following the rules below:

Boxes cannot be stacked or split Boxes can be pushed into the warehouse from both sides (left to right, right to left) If the height of a room in the warehouse is less than the height of the current chest, the current chest can only stay in front of that room If a room is already occupied by a chest, subsequent chests cannot be pushed in the same direction You can adjust the order in which the chests are pushed in

Returns the maximum number of chests that can be pushed into this warehouse.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 boxes.length,warehouse.length 1051 \leq boxes.length, warehouse.length \leq 10^51 boxes.length,warehouse.length 105 1 boxes[i],warehouse[i] 1091 \leq boxes[i], warehouse[i] \leq 10^91 boxes[i],warehouse[i] 109ExampleExample 1: Input: boxes = [1,2,2,3,4] warehouse = [3,4,1,2] Output: 4 Explanation The following diagram shows how to do this 1. push a box of height 1 in any direction into warehouse[2]. 2. push a box of height 2 from right to left into warehouse[3]. 2. push a box of height 2 into warehouse[3] from right to left 3. push a box of height 2 into warehouse[1] from left to right 4. push the box with height 3 into warehouse[0] from left to right Example 2: Input: boxes = [4,3,4,1] warehouse = [5,3,1,3,4] Output: 4 Explanation: The following diagram shows how to do this 1. push a box of height 1 in any direction into warehouse[2]. 2. push a box of height 2 from right to left into warehouse[3]. 2. push a box of height 3 into warehouse[1] or warehouse[3] from any direction 3. push a box of height 4 from each side into warehouse[0] and warehouse[4]. ==

Problem Number: 1713 URL: https://www.lintcode.com/problem/3691 Title: Dot Product of Two Sparse Vectors Problem Description: DescriptionGiven two sparse vectors, compute the product of their quantities (dot product). Implements class SparseVector:

SparseVector(nums) initializes the object with the vector nums dotProduct(vec) computes the product of this vector and the vec vector

Sparse vectors are vectors that have a majority of components 0 and are stored efficiently by designing the SparseVector class. The formula for the product of quantities is as follows: a b= i=0naibi=a1b1+a2b2+... +anbna \cdot b = \sum_{i = 0}^na_ib_i = a_1b_1 + a_2b_2 + ... \ +a_nb_n a b= i=0n aibi=a1b1+a2b2+... +anbnOnly $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 nums.length 1051 \leq nums.length \leq 10^51 nums.length 105 nums1.length==nums2.lengthnums1.length == nums2.lengthnums1.length==nums2.length 0 nums1[i],nums2[i] 10000 \leq nums1[i], nums2[i] \leq 10000 nums1[i],nums2[i] 1000ExampleExample 1: Input: nums1 = [0,0,1,2,0,3] nums2 = [4,0,1,0,0,3] Output: 10 Explanation: v1 = SparseVector(nums1), v2 = SparseVector(nums2) v1.dotProduct(v2) = 0*4 + 0*0 + 1*1 + 2*0 + 0*0 + 3*3 = 10 Example 2: Input: nums1 = [1,0,0,0] nums2 = [0,0,2,0] Output: 0 Explanation: v1 = SparseVector(nums1), v2 = SparseVector(nums2) v1.dotProduct(v2) = 1*0 + 0*0 + 0*2 + 0*0 = 0 ==

Problem Number: 1697 URL: https://www.lintcode.com/problem/3688 Title: Strings Differ by One Character Problem Description: DescriptionGiven a list of strings words, for each string in words, the length is the same. Returns true when there are two strings in words such that they differ by only one character at the same index, otherwise returns false.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 words.length 1041 \leq words.length \leq 10^41 words.length 104 1 words[i].length 1041 \leq words[i].length \leq 10^41 words[i].length 104 No duplicate string exists in words words[i] contains lowercase letters onlyExampleExample 1: Input: words = ["abc", "bcd", "aba"] Output: true Explanation: The strings "abc" and "aba" have a different character at index 2, and are the same at all other indexes. Example 2: Input: words = ["aa", "bb", "cc"] Output: false ==

Problem Number: 1665 URL: https://www.lintcode.com/problem/3686 Title: Diameter of N-Ary Tree Problem Description: DescriptionGiven the root node root of an N-ary tree, return the length of the diameter of the tree. The diameter of an N-ary tree is the length of the longest path between any two nodes in the tree, where this path does not necessarily pass through the root node.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 The depth of the N-tree is less than or equal to 1000. Total number of nodes between [0, 104][\ 0, \ 10^4\ ][0, 104]ExampleExample 1: Input: {1,3,2,4#2#3,5,6#4#5#6} Output: 3 Explanation: The path of the red line is shown below.

Example 2: Input: {1,2,3,4,5#2,6,7#3#4#5#6,8#7,9#8,10#9,11#10#11} Output: 6 Explanation: The path of the red line is shown below.

==

Problem Number: 1609 URL: https://www.lintcode.com/problem/3627 Title: Find All The Lonely Nodes Problem Description: DescriptionIn a binary tree, a lonely node is a node that is the only child of its parent node. The root of the tree is not lonely because it does not have a parent node. Given the root of a binary tree, return an array containing the values of all lonely nodes in the tree. Return the list in any order.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 The number of nodes in the tree is in the range [1,1000][1, 1000][1,1000]. 1<=Node.val<=1061 <= Node.val <= 1061<=Node.val<=106ExampleExample 1 Input root = [1,2,3,#,4] Output [4] Explanation Node 1 is the root and is not lonely. Node 2 and 3 have the same parent 1 and are not lonely. Example 2 Input root = [7,1,4,6,#,5,3,#,#,#,#,#,2] Output [6,2] Explanation Node 7 is the root and is not lonely. Node 1 and 4 have the same parent 7 and are not lonely. Node 5 and 3 have the same parent 4 and are not lonely. Notice that order doesn't matter, [2,6] is also an acceptable answer. Example 3 Input root = [11,99,88,77,null,null,66,55,null,null,44,33,null,null,22] Output [77,55,33,66,44,22] Explanation Node 11 is the root. Node 99 and 88 have the same parent 11. All the other nodes are lonely nodes. Example 4 Input root = [197] Output [] ==

Problem Number: 1391 URL: https://www.lintcode.com/problem/3642 Title: Counting Elements Problem Description: DescriptionThere exists an array arr. You need to count the elements of the array. If a number x exists in arr, and x + 1 also exists in arr, then increment the count by one. Returns the number of elements that meet the requirements.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 The same elements need to be counted separately.ExampleExample 1 Input: [1, 2, 3] Output: 2 Explanation: Both 1 and 2 meet the requirement because both 2(1+1) and 3(2+1) are present in arr. Example 2 Input: [1, 1, 2, 3] Output: 3 Explanation: 1 and 2 both meet the requirement, but since there are two 1s, the result is 2(1) + 1(2) = 3. Example 3 Input: [1, 3, 5] Output: 0 Explanation: No valid element exists. ==

Problem Number: 1345 URL: https://www.lintcode.com/problem/3645 Title: Perform String Shifts Problem Description: DescriptionIn this problem, there is a string s that contains only lowercase characters. Now the string needs to be shifted several times. Each of these shift can be represented by an integer array of length 2. The first element of the array indicates the moving direction of the operation, where 0 means left shift and 1 means right shift. The two parameters of the array represent the number of bits to move.

Move operation: Shifting 1 bit to the left means splicing the first element of the string s to the end of the string, shifting 2 bits to the right means splicing the last 2 elements of the string to the beginning of the string.

Returns the string after several shifts.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: abc [[0, 1], [1, 2]] Output: cab Explanation:

First operation shift left one bit => "bca" The second operation shifts right two places => "cab"

Example 2 Input: abc [[0, 1], [1, 2], [0, 1]] Output: abc Example 3 Input: abc [[1, 3], [0, 3]] Output: abc ==

Problem Number: 1199 URL: https://www.lintcode.com/problem/3640 Title: Hexspeak Problem Description: DescriptionIn this question, you need to check whether the hexadecimal representation of the positive integer n can be converted into hexspeak.

Hexspeak: If the hexadecimal representation of a number is only composed of alphabet {a, b, c, d, e, f} or numbers {0, 1}, then the number can be converted to the hexspeak Conversion rules: Convert the number 0 to the character O, 1 to the character I, and capitalize all other alphabet.

If n can be converted to hexspeak, return its string, otherwise return "ERROR".Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: 7950 Output: IFOA Example 2 Input: 3786 Output: ECA Example 3 Input: 9 Output: ERROR ==

Problem Number: 1192 URL: https://www.lintcode.com/problem/1817 Title: Divide Chocolate Problem Description: DescriptionYou have one chocolate bar that consists of some chunks. Each chunk has its own sweetness given by the array sweetness. You're going to share this chocolate with K friends, so you need to cut K times to get K + 1 pieces, each of which is made up of a series of small pieces. Being generous, you will eat the piece with the minimum total sweetness and give the other pieces to your friends. Find the maximum total sweetness of the piece you can get by cutting the chocolate bar optimally. Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 0 <= K < sweetness.length <= 10^4 1 <= sweetness[i] <= 10^5 ExampleExample 1:

Input: sweetness = [1,2,3,4,5,6,7,8,9], K = 5 Output: 6 Explanation: You can divide the chocolate to [1,2,3], [4,5], [6], [7], [8], [9] Example 2:

Input: sweetness = [5,6,7,8,9,1,2,3,4], K = 8 Output: 1 Explanation: There is only one way to cut the bar into 9 pieces. Example 3:

Input: sweetness = [1,2,2,1,2,2,1,2,2], K = 2 Output: 5 Explanation: You can divide the chocolate to [1,2,2], [1,2,2], [1,2,2] == Problem Number: 1190 URL: https://www.lintcode.com/problem/3669 Title: Smallest Common Region Problem Description: DescriptionThere is a list of regions, where the first region of each sub-list contains all the other regions in the list. That is, if region X contains region Y, then region X is larger than region Y. Given two regions region1 and region2, find the minimum region that contains both regions. If r1 in the list of regions contains r2 and r3, it is guaranteed that there will be no containment relationship between r2 and r3. It is also guaranteed that the least common region must exist.regions[i][j], region1, and region2 consist of English letters. 1<=regions[i][j].length,region1.length,region2.length<=201 <= regions[i][j].length, region1.length, region2.length <= 201<=regions[i][j].length,region1.length,region2.length<=20 2<=regions[i].length<=202 <= regions[i].length <= 202<=regions[i].length<=20 2<=regions.length<=1042 <= regions.length <= 10^42<=regions.length<=104 region1 region2region1 \neq region2region1 =region2ExampleExample 1 Input regions = [["Earth","North America","South America"], ["North America","United States","Canada"], ["United States","New York","Boston"], ["Canada","Ontario","Quebec"], ["South America","Brazil"]] region1 = "Quebec" region2 = "New York" Output "North America" Example 2 Input regions = [["Earth","North America","South America"], ["North America","United States","Canada"], ["United States","New York","Boston"]] region1 = "New York" region2 = "Canada" Output "North America" Example 3 Input regions = [["Earth","North America","South America"], ["North America","United States","Canada"], ["United States","New York","Boston"], ["South America","Brazil"]] region1 = "Boston" region2 = "Brazil" Output "Earth" ==

Problem Number: 1189 URL: https://www.lintcode.com/problem/3667 Title: Encode Number Problem Description: DescriptionYou are given a non-negative integer num and you need to return its encrypted string. The process of encryption involves transforming an integer with some unknown function, which you need to deduce from the following table: numen-

code(num)0""1"0"2"1"3"00"4"01"5"10"6"11"7"000"Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 0<=num<=1090 <= num <= 10^90<=num<=109ExampleExample 1 Input num = 10 Output "011" Example 2 Input num = 23 Output "1000" Exampe 3 Input num = 107 Output "101100" ==
Problem Number: 1175 URL: https://www.lintcode.com/problem/3636 Title: Array Transformation Problem Description: DescriptionThere exists an array arr. You need to transform the elements in the array except the head and tail elements until the array can no longer be transformed. The transformation rules are as follows:

If a number is smaller than its two adjacent numbers at the same time, then this number needs to be increased by 1 If a number is greater than its two adjacent numbers at the same time, then this number needs to be reduced by 1

Finally returns the final array which cannot be transformed.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: [1, 2, 1] Output: [1, 1, 1] Example 2 Input: [5, 12, 3, 8, 9] Output: [5,7,8,8,9] Example 3 Input: [9, 6, 7, 10, 13, 14] Output: [9,7,7,10,13,14] ==
Problem Number: 1165 URL: https://www.lintcode.com/problem/3653 Title: Meeting Scheduler Problem Description: DescriptionGiven the availability schedules slots1 and slots2 of two people, and the expected duration of the meeting, you are asked to schedule the earliest and appropriate meeting time in the interval for them. The format of an availability schedules is an Interval consisting of a start time and an end time, i.e., (start, end), which means that it starts at start and ends at end. If the required meeting time is not met, return to the interval (-1, -1).Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1<=slots1.length,slots2.length<=1041 <= slots1.length, slots2.length <= 10^41<=slots1.length,slots2.length<=104 slots1[i].length,slots2[i].length==2slots1[i].length, slots2[i].length == 2slots1[i].length,slots2[i].length==2 slots1[i].start<slots1[i].endslots1[i].start < slots1[i].endslots1[i].start<slots1[i].end slots2[i].start<slots2[i].endslots2[i].start < slots2[i].endslots2[i].start<slots2[i].end 0<=slots1[i][j],slots2[i][j]<=1090 <= slots1[i][j], slots2[i][j] <= 10^90<=slots1[i][j],slots2[i][j]<=109 1<=duration<=1061 <= duration <= 10^61<=duration<=106ExampleExample 1 Input slots1 = [(10,50),(60,120),(140,210)] slots2 = [(0,15),(60,70)] duration = 8 Output (60,68) Example 2 Input slots1 = [(10,50),(60,120),(140,210)] slots2 = [(0,15),(60,70)] duration = 12 Output (-1,-1) ==
Problem Number: 1164 URL: https://www.lintcode.com/problem/3633 Title: Missing Number In Arithmetic Progression Problem Description: DescriptionThere exists an array arr whose elements form an arithmetic progression.

Arithmetic progression: arr[n]=arr[1]+(n−1)darr[n] = arr[1] + (n - 1) darr[n]=arr[1]+(n−1)d, that is, arr[n+1]−a[n]=darr[n + 1] - a[n] =

darr[n+1]−a[n]=d. where ddd is a constant.

In this problem, the arithmetic progression arr has one element removed from the middle (neither the first element nor the last element), and you need to find and return the removed element.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 3<=arr.length<=100 3 <= arr.length <= 1003<=arr.length<=100ExampleExample 1 Input [2, 5, 8, 14] Output 11 Example 2 Input [1, 2, 4, 5] Output 3 Example 3 Input [5, 0, -10, -15] Output -5 Example 4 Input: [10, 8, 7] Output: 9 Explanation The original array was [10, 9, 8, 7] == Problem Number: 1149 URL: https://www.lintcode.com/problem/3629 Title: Intersection of Three Sorted Arrays Problem Description: DescriptionGiven three strictly increasing integer arrays arr1, arr2 and arr3, return the intersection of these three arrays.ExampleExample 1 Input: [1, 2, 3] [2, 3, 4] [3, 4, 5] Output: [3] Explanation: Only 3 exists in all three arrays. Example 2 Input: [1] [2] [3] Output: [] Example 3 Input: [1, 2, 4, 8] [2, 4, 8] [1, 3, 8] Output: [8] == Problem Number: 1143 URL: https://www.lintcode.com/problem/3684 Title: Find Smallest Common Element in All Rows Problem Description: DescriptionThere exists a m×nm \times nm×n matrix mat with strictly increasing elements in each row. You need to return the smallest common element among all rows. Returns -1 if not present.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 m==mat.lengthm == mat.lengthm==mat.length n==mat[i].lengthn == mat[i].lengthn==mat[i].length 1 m,n 5001 \leq m, n \leq 5001 m,n 500 1 mat[i][j] 1041 \leq mat[i][j] \leq 10^41 mat[i][j] 104ExampleExample 1 Input: [[1,2,3],[2,3,4],[3,4,5]] Output: 3 Example 2 Input: [[1,2,3],[2,3,4],[3,4,5],[4,5,6]] Output: -1 == Problem Number: 1141 URL: https://www.lintcode.com/problem/3626 Title: How Many Apples Can You Put into the Basket Problem Description: DescriptionIn this problem, you have a basket that can be used to hold apples, and the maximum capacity (weight) of the basket is 5000. Now there is a bunch of apples, where the weight of the ith apple is weights[i]. You can pick any of them and put them in the basket, and return the maximum number of apples you can put in.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: [5000, 1000, 2000, 3000] Output: 2 Explanation: Can hold up to 2 apples. For example, select [1000, 2000] or [2000, 3000]. Example 2 Input: [5000, 10000] Output: 1 Example 3 Input: [10000] Output: 0 == Problem Number: 1134 URL: https://www.lintcode.com/problem/3679 Title: Shortest Distance to Target Color Problem Description: DescriptionIn this question, there is a colors array containing only three colors 1, 2, and

3. In addition, there is a two-dimensional integer array of queries. Each sub-array contains two parameters, namely i, c. Where i represents the subscript of the array, and c represents the color value. Each sub-array is a shortest distance query. This query needs to get the closest distance between the element with subscript i in colors and the target color c. Returns -1 if there is no target color, and the closest distance is 0 if the subscript color is the same as the target color. You need to query all the results in queries and store them in an array to return.ExampleExample 1 Input: [1,1,2,1,3] [[1,3],[2,3]] Output: [3,2] Example 2 Input: [1,2] [[1,3]] Output: [-1] ==

Problem Number: 1132 URL: https://www.lintcode.com/problem/3680 Title: Before and After Puzzle Problem Description: DescriptionGiven a list of phrases, phrases, generate a list of spliced new phrases according to the rules of forward and backward splicing.

A phrase is a string of words containing only lowercase letters, no spaces at the beginning or end, and only one space between words, and spaces.

The rules for specifying front and back splices are as follows:

When the last word of the first phrase and the first word of the second phrase are identical, the two phrases are combined into a new phrase When splicing, only one word remains in the spliced area, e.g. "a b c" and "c d e" will be spliced as a b c d e Only two phrases can be combined per splice The same phrase can participate in multiple splices, but the spliced new phrase cannot participate in the splice A phrase cannot be self-spliced with itself

Returns a list of all generated new phrases in dictionary order, with duplicates removed from the list.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 phrases.length 1001 \leq phrases.length \leq 1001 phrases.length 100 1 phrases[i].length 1001 \leq phrases[i].length \leq 1001 phrases[i].length 100ExampleExample 1: Input: phrases = ["lint code", "code everyday"] Output: ["lint code everyday"] Example 2: Input: pharses = ["a party with single", "single dance", "a man on a party", "mission within mission"] Output: ["a party with single dance"] Example 3: Input: pharses = ["lintcode", "jiuzhang", "lintcode", "jiuzhang"] Output: ["jiuzhang", "lintcode"] ==

Problem Number: 1131 URL: https://www.lintcode.com/problem/3624 Title: Count Substrings with Only One Distinct Letter Problem Description: DescriptionIn this question, you need to calculate the number of substrings that contain only one type of character among all the substrings of the string s.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 Returns 0 if the string length is 0.ExampleExample 1: Input: aabb Output: 6 Explanation: In this string, the substrings "a", "a", "b", "b", "aa", "bb" are all substrings containing only one kind of characters. Example 2: Input: abcd Output: 4 Example 3: Input: aaaa Output: 10

==
Problem Number: 1126 URL: https://www.lintcode.com/problem/1872 Title: Minimum Cost to Connect Sticks Problem Description: DescriptionIn order to decorate your new house, you need to process some sticks with positive integer length. You can connect any two sticks of lengths X and Y into one stick by paying a cost of X + Y. Due to the construction needs, you must connect all the bars into one. Return the minimum cost of connecting all the given sticks into one stick in this way. Please note that you can choose any order of sticks connectionOnly $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 sticks.length 1041 \leq sticks.length \leq 10^41 sticks.length 104 1 sticks[i] 1041 \leq sticks[i] \leq 10^41 sticks[i] 104 ExampleExample 1: Input: [2,4,3] Output: 14 Explanation: First connect 2 and 3 to 5 and cost 5; then connect 5 and 4 to 9; total cost is 14 Example 2: Input: [1,8,3,5] Output: 30
==
Problem Number: 1125 URL: https://www.lintcode.com/problem/3677 Title: Design File System Problem Description: DescriptionIn this problem, you need to design a file system that can create a new path and bind a value. Each of these paths begins with a separator /, and there must be several lowercase letters after each separator. For example, /lint and /lint/code are valid paths, while an empty string and / are not. You need to implement the following method:

bool createPath(string path, int value): Determine whether path can be created, if yes, create and bind the corresponding value, and return true at the same time, otherwise return false.

int get(string path): If the path path exists, return the value associated with path, otherwise return -1.

Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 Paths cannot be recreated to modify bound values The premise of creating a path is that the preceding paths have been created in advance. For example, the initial path is /lint, which can be created, but /lint/code cannot because we must create /lint first.ExampleExample 1 Input: createPath("/a", 1) createPath("/a/b", 2) get("/a") get("/a/b") Output: true true 1 2 Example 2 Input: createPath("/a", 1) createPath("/a/b/c", 2) get("/a") get("/a/b") Output: true false 1 -1
==
Problem Number: 1123 URL: https://www.lintcode.com/problem/3638 Title: Single-Row Keyboard Problem Description: DescriptionA custom string keyboard keyboard of fixed length 26, consisting of 26 lowercase letters all aligned on one line. The custom keyboard has index subscripts from 0 to 25. Initially, your finger is at index subscript 0, and each time you type a character, you need to move your finger to the index of the corresponding character in the keyboard keyboard, and the number of moves required to move from index i to index j

is |i - j|. Now given a word string word, calculate how many times you need to move it to type the word.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 keyboard.length==26keyboard.length == 26keyboard.length==26 1 word.length 1051 \leq word.length \leq 10^51 word.length 105 keyboard contains only lowercase letters, and each lowercase letter has one and only one word contains only lowercase lettersExampleExample 1: Input: keyboard = "abcdefghijklmnopqrstuvwxyz" word = "abc" Output: 2 Explanation: Start position is "a", no need to move "a" -> "b" move 1 time "b" -> "c" move 1 time Total move 2 times Example 2: Input: keyboard = "gsdwqxfavuhbimkeycptnrzljo" word = "lintcode" Output: 89 ==

Problem Number: 1108 URL: https://www.lintcode.com/problem/3676 Title: Analyze User Website Visit Pattern Problem Description: DescriptionNow given two arrays of strings username and website and an array of integers timestamp, and given that all three arrays are of the same length, for index i, the elements of the three arrays form a triple (username[i], timestamp[i], website[i]) indicating that the user username[i] visited the website website[i] at timestamp[i]. Access behavior is a list of websites that the user has visited in order, and duplicate elements may appear in the list.

If the access behavior is ["course", "problem", "collection"], it means that a user visited "course", then "problem", and then "collection" in order of timestamp If the access behavior is ["group", "problem", "group"], it means that a user accessed "group", then "problem", then "group", in order of timestamp If the access behavior is ["lintcode", "lintcode", "lintcode"], it means that the user accessed "lintcode" three times in a row, on different timestamp

For all users in username, return the most frequent access behavior. If there are multiple accesses with the same number of occurrences, return the access with the lowest dictionary order.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 3 username.length 503 \leq username.length \leq 503 username.length 50 1 username[i].length 101 \leq username[i].length \leq 101 username[i].length 10 username.length==timestamp.length==website.lengthusername.length == timestamp.length == website.lengthusername.length==timestamp.length==website.length 1 timestamp[i] 1091 \leq timestamp[i] \leq 10^91 timestamp[i] 109 1 website[i].length 101 \leq website[i].length \leq 101 website[i].length 10 username[i] and website[i] contains only lowercase characters The test data guarantees that all users in username have visited the website at least three timesExampleExample 1: Input: username = ["alex","alex","alex","ken","ken","ken","ken","mary","mary","mary"] timestamp = [1,2,3,4,5,6,7,8,9,10] website = ["lintcode","course","problem","lintcode","group","collection","lin Output: ["lintcode","course","problem"] Explanation: There are three users: alex, ken, mary, and their access behavior are as follows: alex: ["alex","lintcode",1], ["alex","course",2], ["alex","problem",3] ken: ["ken","lintcode",4], ["ken","group",5], ["ken","collection",6], ["ken","lintcode",7] mary: ["mary","lintcode",8], ["mary","course",9], ["mary","problem",10] For the three users above, a total of five user access behaviors occurred: ["lintcode","course","problem"] appeared 2 times (alex and mary) ["lintcode","group","collection"] appeared 1 time (ken) ["lintcode","group","lintcode"] appeared 1 time (ken) ["lintcode","collection","lintcode"] appeared 1 time (ken) ["group","collection","lintcode"] appeared 1 time (ken) Example 2: Input: username = ["usera","usera","usera","userb","userb","userb"] timestamp = [1,2,3,4,5,6] website = ["a","b","c","a","b","a"] Output: ["a","b","a"] ==

Problem Number: 1107 URL: https://www.lintcode.com/problem/3675 Title: Minimum Swaps to Group All 1's Together Problem Description: DescriptionIn this problem there is a binary array (contains only 0 and 1). You can swap arbitrary elements of the array. Now you need to combine all 1s in the array. Returns the minimum number of swaps required for a combination.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input [1,0,1,0,1] Explanation: [1,1,1,0,0] or [0,0,1,1,1] only need one swap. Output: 1 Example 2 Input: [1,0,1,0,1,0,1,1,0,1] Output: 2 Explanation: [0,0,0,0,1,1,1,1,1,1] requires only two swaps. ==

Problem Number: 1101 URL: https://www.lintcode.com/problem/3673 Title: Parallel Courses Problem Description: DescriptionIn this problem, there are n courses. They are numbered 1 to n. There is also a two-dimensional array relations. Inside each array contains two courses. The first course is a pre-requisite for the second course. That is, you must learn the first course in the array before learning the second course in the array. You may study any number of courses in a semester, provided you have completed all prerequisites for that course in the previous semester. Return minimum semester of full coursework, or -1 if full coursework cannot be completed.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: 4 [[1,2],[2,3],[2,4]] Output: 3 Explanation:

Semester 1: Study 1 Semester 2: Study 2 Semester 3: Study 3, 4

Example 2 Input: 4 [[1,2],[2,3],[2,4],[3,4]] Output: 4 Example 3 Input: 2 [[1,2],[2,1]] Output: -1 ==

Problem Number: 1100 URL: https://www.lintcode.com/problem/3672 Title: Connecting Cities With Minimum Cost Problem Description: DescriptionThere are n cities in this question, and their numbers range from 1 to n. At the same time, there is a connections array and connections[i]=[ai,bi,ci]connections[i] = [a_i, b_i, c_i]connections[i]=[ai,bi,ci], which means that the cost of connecting cities aia_iai and bib_ibi is cic_ici. Please return the minimum cost required to connect all cities. If all cities cannot be connected, return -1.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: 3 [[1,2,1], [2,3,2], [1,3,3]] Ouput: 3 Explanation: Choose [1,2,1] and [2,3,2] to connect all n cities. At this time, the cost is the least, which is 3. Example 2 Input: 3 [[1,2,1]] Output: -1 Explanation: Unable to connect all cities according to connections. ==

Problem Number: 1098 URL: https://www.lintcode.com/problem/3634 Title: Largest Unique Number Problem Description: DescriptionGiven an array of integers nums, returns the largest and only once occurring integer in the array. If it does not exist, return -1.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 0 nums.length 50000 \leq nums.length \leq 50000 nums.length 5000 0 nums[i] 10000 \leq nums[i] \leq 10000 nums[i] 1000ExampleExample 1: Input: nums = [3,5,1,7,9,1,9] Output: 7 The integer 9 appears twice in nums and only once and the largest is the integer 7 Example 2: Input: nums = [1,9,2,2,9,1] Output: -1 There are no integers that occur only once in nums ==

Problem Number: 1085 URL: https://www.lintcode.com/problem/3670 Title: The Earliest Moment When Everyone Become Friends Problem Description: DescriptionIn a social circle, there are n people, numbered from 0 to n - 1. Now there is a list of logs logs, where logs[i] = [time, x, y] indicates that x and y became friends with each other at time. Friendship is mutual and transferable. That is:

Mutual: when a and b are friends, then b has a among his friends Transferable: when a and b are friends, and b has c as a friend, then a and c will also be friends and know each other

Return the earliest time when everyone in this circle knew each other. If the earliest time is not found, -1 is returned.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 2 n 1002 \leq n \leq 1002 n 100 1 logs.length 1041 \leq logs.length \leq 10^41 logs.length 104 logs[1] !=logs[2]logs[1] \ != logs[2]logs[1] !=logs[2] In the logs element, there will be no duplicate elements for a given time timeExampleExample 1: Input: logs = [[20220101,0,1],[20220109,3,4],[20220304,0,4],[20220305,0,3],[20220404,2,4]] n = 6 Output: 20220404 Explanation: time = 20220101 0 and 1 have become friends with [0,1], [2], [3], [4] time = 20220109 3 and 4 have become friends with [0,1], [2], [3,4] time = 20220304 0 and 4 have become friends with [0,1,3,4], [2] time = 20220305 0 and 3 already friends time = 20220404 2 and 4 have become friends with [0,1,2,3,4], everyone konws each other Example 2: Input: logs = [[7,3,1],[3,0,3],[2,0,1],[1,1,2],[5,3,2]] n = 4 Output: 3 ==

Problem Number: 1082 URL: https://www.lintcode.com/problem/3623 Title: Sum of Digits in the Minimum Number Problem Description: DescriptionGiven a non-empty positive integer array nums, finds the smallest value in the array nums. If the sum of the digits of the smallest value is even, True is returned, otherwise, False is returned.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes

Twitter for more information WeChat ID jiuzhang104 1 nums.length 1001 \leq nums.length \leq 1001 nums.length 100 1 nums[i] 1091 \leq nums[i] \leq 10^91 nums[i] 109ExampleExample 1: Input: nums = [20,10,15,19] Output: False Explanation: The smallest element is 10 The sum of the digits of this element is 1 + 0 = 1, which is an odd number Example 2: Input: nums = [444,555,333,222] Output: True Explanation: The smallest element is 222 The sum of the digits of this element is 2 + 2 + 2 = 6, which is an even number ==

Problem Number: 1075 URL: https://www.lintcode.com/problem/3625 Title: Index Pairs of a String Problem Description: DescriptionGiven a text string and words (a list of strings), return all index pairs [i, j] so that the substring text[i]...text[j] is in the list of words.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 All strings contains only lowercase English letters. It's guaranteed that all strings in words are different. Return the pairs [i,j] in sorted order (i.e. sort them by their first coordinate in case of ties sort them by their second coordinate). 1<=text.length<=1001 <= text.length <= 1001<=text.length<=100 1<=words.length<=201 <= words.length <= 201<=words.length<=20 1<=words[i].length<=501 <= words[i].length <= 501<=words[i].length<=50ExampleExample 1 Input text = "thestoryofleetcodeandme" words = ["story","fleet","leetcode"] Output [[3,7],[9,13],[10,17]] Example 2 Input text = "ababa" words = ["aba","ab"] Output [[0,1],[0,2],[2,3],[2,4]] Explanation Matches can overlap, see "aba" is found in [0,2] and [2,4]. ==

Problem Number: 1067 URL: https://www.lintcode.com/problem/3666 Title: Campus Bikes II Problem Description: DescriptionThere are n workers and m bicycles in a campus represented by a 2D grid, n <= m. The locations of all workers and bicycles are represented by 2D coordinates on the grid. Given the Manhattan distance between two points P1P_1P1 and P2P_2P2 is Manhattan(P1,P2)= P1.x−P2.x + P1.y−P2.y Manhattan(P_1, P_2) = |P_1.x - P_2.x| + |P_1.y - P_2.y|\ Manhattan(P1,P2)= P1.x−P2.x + P1.y−P2.y . We need to assign a bicycle to each worker. such that the sum of the Manhattan distances and the sum of the Manhattan distances between all workers and their assigned bikes is minimized. For all allocation scenarios, return the smallest possible sum of Manhattan distances and sums between all workers and their assigned bikes.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 0 workers[i][j],bikes[i][j]<10000 \leq workers[i][j], bikes[i][j] < 10000 workers[i][j],bikes[i][j]<1000 1 workers.length bikes.length 101 \leq workers.length \leq bikes.length \leq 101 workers.length bikes.length 10 1 n m 101 \leq n \leq m \leq 101 n m 10 All workers and bikes are in different locations.ExampleExample 1: Input: workers = [[0,0],[2,1]] bikes = [[1,2],[3,3]] Output: 6 Explanation: Worker 0 is assigned to Bicycle 0 and Worker 1 is assigned to Bicycle 1. The Manhattan distance between worker 0 and bike 0 is: |1 - 0| + |2 - 0| = 3 The Manhattan distance between Worker 1 and bike 1 is: |3 - 2| + |3 - 1| = 3 This solution minimizes the sum of the Manhattan distances and returns 6

Example 2: Input: workers = [[0,0],[1,1],[2,0]] bikes = [[1,0],[2,2],[2,1]] Output: 4 Explanation: Worker 0 is first assigned to bike 0, worker 1 is assigned to bike 1 (or bike 2), and worker 2 will be assigned to bike 2 (or bike 1) The Manhattan distance between worker 0 and bike 0 is: |1 - 0| + |0 - 0| = 1 The Manhattan distance between Worker 1 and bike 1 is: |2 - 1| + |2 - 1| = 2 The Manhattan distance between worker 2 and bike 2 is: |2 - 2| + |1 - 0| = 1 This solution minimizes the sum of the Manhattan distances and returns 4.

==
Problem Number: 1066 URL: https://www.lintcode.com/problem/3621 Title: Fixed Point Problem Description: DescriptionGiven a non-empty, strictly ascending array nums, if there exists an index i that satisfies nums[i] == i, then i is the fixed point of the array nums. Returns the minimum fixed point of the array nums, or -1 if it does not exist.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 nums does not contain duplicate elements 1 nums.length<1041 \leq nums.length < 10^41 nums.length<104 −109 nums[i] 109-10^9 \leq nums[i] \leq 10^9−109 nums[i] 109ExampleExample 1: Input: nums = [-3,-1,0,1,3,4,6] Output 6 Explanation: nums[0] = -3, nums[1] = -1, ... , nums[5] = 4, nums[6] = 6 The output is therefore 6 Example 2 Input: nums = [0,1,3,5,7] Output: 0 Explanation: nums[0] = 0, nums[1] = 1 0 is the minimum fixed point, so it returns 0 Example 3: Input: nums = [-1,0,1,2] Output: -1 Explanation: No fixed points, return -1
==
Problem Number: 1060 URL: https://www.lintcode.com/problem/3664 Title: Longest Repeating Substring Problem Description: DescriptionGiven a string s, find the longest and recurring substring and output its length, the substrings can have overlapping parts but not completely overlapping, if there is no repeating substring, return 0.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 str.length 10001 \leq str.length \leq 10001 str.length 1000 String s contains only lowercase lettersExampleExample 1: Input: s = "aaaa" Output: 3 Explanation: The longest and recurring substring is "aaa", which is repeated 2 times in s Example 2: Input: s = "aabcaabdaab" Output: 3 Explanation: The longest and recurring substring is "aab", which is repeated 3 times in s
==
Problem Number: 1059 URL: https://www.lintcode.com/problem/3661 Title: Missing Element in Sorted Array Problem Description: DescriptionNow there is an array of integers nums in strictly ascending order, and there are no identical elements in nums. Given an integer k, find and return the kth missing numeric element in nums, starting from the left.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 nums.length 1051 \leq nums.length \leq 10^51 nums.length 105 1 nums[i] 1071 \leq nums[i]

\leq 10^71 nums[i] 107 1 k 1081 \leq k \leq 10^81 k 108ExampleExample 1: Input: nums = [1,3,5,7,9] k = 1 Output: 2 Expplanation: The first missing digit is 2 Example 2: Input: nums = [1,3,5,7,9] k = 4 Output: 8 Explanation: The missing digits are [2,4,6,8,...]. and the fourth missing digit is 8. Example 3: Input: nums = [2,3,4,5,7] k = 4 Output: 10 Explanation: The missing digits are [6,8,9,10,...]. The fourth missing digit is 10. ==

Problem Number: 1053 URL: https://www.lintcode.com/problem/3655 Title: Minimize Rounding Error to Meet Target Problem Description: DescriptionGiven an array of floating point numbers nums and a target value target, for each element of the array nums [x1, x2, ... , xn][x_1,\ x_2,\ ...\ ,\ x_n][x1, x2, ... , xn] is rounded down floor(xi)floor(x_i)floor(xi) or up ceil(xi)ceil(x_i)ceil(xi) so that the new array [X1, X2, ... , Xn][X_1,\ X_2,\ ... \ ,\ X_n][X1, X2, ... , Xn] sum equals the target value target. If the elements of the given array nums cannot be rounded so that the sum equals target, then "-1" is returned in string format, otherwise, the smallest adjusted sum is returned in string format, retaining two decimal places. The adjusted sum is defined as xi−Xi |x_i - X_i| xi−Xi , for example:

Rounding up a floating point number 1.2 to get 2.0 gives an adjusted sum of 0.8. Rounding down to a floating point number 1.2 gives 1.0, which sums to 0.2. Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 nums.length 10001 \leq nums.length \leq 10001 nums.length 1000 nums[i] is between [0.00, 1000.00] and has no more than 2 decimal places The returned data type is a string typeExampleExample 1: Input: nums = [1.2, 1.7, 2.3] target = 5 Output: "0.80" Explanation: The result of rounding to get the minimum adjusted sum is [1.0, 2.0, 2.0] The adjusted sum is |1.2 - 1.0| + |1.7 - 2.0| + |2.3 - 2.0| = 0.8 Example 2: Input: nums = [1.2, 2.3, 3.4] target = 10 Output: "-1" Explanation: Can't get the sum of the nums to 10 by rounding up the numbers ==

Problem Number: 1051 URL: https://www.lintcode.com/problem/3652 Title: Shortest Way to Form String Problem Description: DescriptionFor any non-empty string, a subsequence of the string can be constructed by selectively deleting some of its characters, e.g., abcde can be obtained by deleting the characters a and c to obtain the subsequence bde, but aebc cannot be constructed by deleting the characters. Now given a string s and a target string target, if the target string target can be formed from s by constructing a subsequence in s and splicing it in series, return the minimum number of subsequences that form the target string target, or -1 if the target string cannot be spliced.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 1 s.length,target.length 50001 \leq s.length, target.length \leq 50001 s.length,target.length 5000 Only lowercase letters are included in s and target.ExampleExample 1: Input: s = "abc"

target = "abcbc" Output: 2 Explanation: The target string "abcbc" can be formed by concatenating the subsequences of "abc": "abc" and "bc". Example 2: Input: s = "abcde" target = "abcdf" Output: -1 Explanation: The presence of "f" in the target string cannot be obtained from the subsequence of s
==
Problem Number: 604 URL: https://www.lintcode.com/problem/3647 Title: Design Compressed String Iterator Problem Description: DescriptionIn this problem, you need to design a compressed string iterator. This iterator class needs to contain a constructor that accepts a compressed string.

Compressed String: It consists of several pairs of characters + numbers. The numbers in each pair represent the number of times that character appears there. For example: the original string represented by "a1b2c3" is "abbccc".

You also need to design the next() and hasNext() methods. Its method is defined as follows:

next(): get the next character of original string hasNext() determines whether the original string has the next character Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 The number after the character must be greater than 0. The string contains only uppercase and lowercase characters, numbers. If no character exists when the next() function is called, '#' is returned.ExampleExample 1 Input: "a1b1" next() hasNext() next() hasNext() Output: 'a' true 'b' false Explanation: The original string represented by the compressed string "a1b1" is "ab". So the first call to next() gets the first character 'a', and the next character exists and is 'b', so the first hasNext() returns true. The second next() accesses the last character 'b', and there is no next character when calling hasNext() at this time, so false is returned. Example 2 Input: "a1b2c3" next() next() next() next() next() next() hasNext() Output: 'a' 'b' 'b' 'c' 'c' 'c' false Example 3 Input: "L1i1n1t1C1o1d1e1" next() next() next() next() hasNext() next() next() next() next() hasNext() Output: 'L' 'i' 'n' 't' true 'C' 'o' 'd' 'e' false
==
Problem Number: 509 URL: https://www.lintcode.com/problem/3665 Title: Inorder Successor in BST II Problem Description: DescriptionThere exists a binary search tree, and the binary tree node holds its parent node parent. In this question, you will get any node of the binary search tree. You need to return the node after this node in the inorder traversal order, or null if there is no successor node.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: {2, 1, 3} 1 Output: 2 Explanation: A binary search tree looks like this: 2 / \ 1 3 Obviously node 1's inorder successor node is 2. Example 2 Input: {2, 1, 3} 3 Output: null
==
Problem Number: 379 URL: https://www.lintcode.com/problem/3659 Title: Design Phone Directory Problem Description: DescriptionIn this question, you

need to design a phone directory management system class. The constructor of the telephone directory management system will receive a variable maxNumbers of integer type, which represents the maximum number of the telephone directory. In addition to the constructor, you also need to design the following functions in this class reasonably:

int get(): Get the next available number in the phone directory, or return -1 if there is no available number bool check(int number): Check if the specified number is available void release(int number): Modify the status of a number to be available Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 The range of available numbers is from 0 to maxNumbers, not including maxNumbers. You need to assign according to the value of the number from small to large.ExampleExample 1 Input: 3 ["get()", "check(0)", "get()", "get()", "release(2)", "check(2)", "get()", "check(2)"] Output: [0, false, 1, 2, null, true, 2, false] Example 2 Input: 0 ["get()", "check(0)", "release(0)"] Output: [-1, false, null] ==

Problem Number: 362 URL: https://www.lintcode.com/problem/3662 Title: Design Hit Counter Problem Description: DescriptionIn this question, you need to design a HitCounter class. In this class, there are the following functions:

HitCounter(): No-argument constructor void hit(int timestamp): Indicates that a tap occurs at the specified time int getHits(int timestamp): Returns the total number of hits within 300 seconds before the specified time

Where timestamp is in seconds.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 You can assume that when the system calls the void hit(int timestamp) and int getHits(int timestamp) methods, it must be called in chronological order, that is, the smaller timestamp is called first. There may be multiple hits at the same time. Hit action takes negligible time. At the same moment (timestamp), getHits() will not happen before hit().ExampleExample 1 Input: ["hit(1)", "hit(2)", "hit(3)", "getHits(3)", "hit(301)", "getHits(301)"] Output: [null, null, null, 3, null, 3] Example 2 Input: ["hit(1)", "hit(1)", "hit(1)", "getHits(2)"] Output: [null, null, null, 3] ==

Problem Number: 359 URL: https://www.lintcode.com/problem/3620 Title: Logger Rate Limiter Problem Description: DescriptionDesign a logger system that receive stream of messages along with its timestamps. A message should be printed if and only if it has not been printed in the last 10 seconds. In the method couldPrintMessage(), given a message and a timestamp in seconds, return true if the message can be printed within the given timestamp, otherwise return false.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 Different messages can be printed at the same moment, but not the same message. At the end of a

message, it is possible to immediately print that message again. The input data will be sorted by timestamp in positive order.ExampleExample 1 Input [[1,foo],[2,bar],[3,foo],[8,bar],[10,foo],[11,foo]] Output True True False False False True Explanation [1,foo] => message "foo" needs to be printed at time 1, when the queue for message "foo" is empty and can be printed, return true [2,bar] => message "bar" needs to be printed at time 2, when the queue for message "bar" is empty and can be printed, return true [3,foo] => message "foo" needs to be printed at time 3, when the queue for message "foo" is not empty (predecessor moment 1 is still in progress), so it cannot be printed, return false [8,bar] => message "bar" needs to be printed at time 8, when the queue for message "bar" is not empty (predecessor moment 2 is still in progress), so it can't be printed, false [10,foo] => message "foo" needs to be printed at time 10, when the queue for message "foo" is not empty (predecessor moment 1 is still in progress), so it cannot be printed, return false [11,foo] => message "foo" needs to be printed at time 11, when the queue for message "foo" is empty (predecessor moment 1 has finished at moment 11), so it can be printed, return true Example 2 Input [[1,foo],[1,bar],[1,bar],[11,foo],[11,bar]] Output True True False True True
==
Problem Number: 353 URL: https://www.lintcode.com/problem/3656 Title: Design Snake Game Problem Description: DescriptionIn this question, you need to design a snake game class SnakeGame.

Snake Game: There exists a snake in a matrix. Food will appear continuously in this matrix, and we need to control the snake to move continuously (through the four directions of up, down, left, and right) to eat the food. after eating food. The length of the snake will increase by one bit. At the same time, the game score will be increased by 1. Afterwards, the next food will be refreshed (the position occupied by the snake will not be refreshed). If the snake hits the wall or eats its own body while eating the food, the game is over.

The function's constructor receives the following parameters:

width: the number of columns in the snake matrix interface height: the number of rows in the snake matrix interface foods: The coordinates of the food, represented by a two-dimensional array, where the first element of each array is the row coordinate, and the second element is the column coordinate

In addition, you need to complete the move() function, which will receive a string indicating the direction. Indicates that the snake needs to move one space in this direction. You need to return the Snake score after the move (initial score is 0). Returns -1 if the game ends after the snake moves.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: 3 2 [[1, 2], [0, 0]] ["r", "d", "r", "u", "l", "l", "l"] Output: [0, 0, 1, 1, 1, 1, 2, -1] Explanation: The initial state is as follows: s | | | | f Among them, s represents the snake, and f represents the current food. After action "r": | s | | | f ... After eating the first food: f | | | s |s ... After eating the second food: s | s | s | | Example 2 Input: 3 2 [[0, 1], [1, 1], [1, 0],

[0, 2]] ["r", "d", "l", "u", "r", "r", "d", "l", "u"] Output: [1, 2, 3, 3, 3, 4, 4, 4, -1]
==

Problem Number: 323 URL: https://www.lintcode.com/problem/3651 Title: Number of Connected Components in an Undirected Graph Problem Description: DescriptionIn this problem, there is an undirected graph with n nodes. There is also an edges array. Where edges[i] = [a, b] means that there is an edge between node a and node b in the graph. You need to return the number of connected components in that graph.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: 3 [[0,1], [0,2]] Output: 1 Example 2 Input: 6 [[0,1], [1,2], [2, 3], [4, 5]] Output: 2
==

Problem Number: 157 URL: https://www.lintcode.com/problem/3622 Title: Read N Characters Given Read4 Problem Description: DescriptionIn this question, you need to design the read() function to read and save the first n characters of the file into buf, and return the length of the read string. You cannot access the file directly, you need to read the file indirectly through the read4() function. Among them, the read4() function reads 4 characters at a time, and if it is called multiple times, it will continue to read from the last read result. Similarly, this function will also return the length of the string actually read. You can see more explanation in the Example.Only $39.9 for the "Twitter Comment System Project Practice" within a limited time of 7 days! WeChat Notes Twitter for more information WeChat ID jiuzhang104 ExampleExample 1 Input: "lintcode" 5 Output: 5 lintc Explanation: The content of file is lintcode, the first call to read4() reads lint, and the second call only needs to read the fifth character (c). So the final read string is lintc with a length of 5. Example 2 Input: "lintcode" 9 Output: 8 lintcode Example 3 Input: "lintcode" 0 Output: 0 "" Note: "" is actually an empty string.
==

Problem Number: 3158 URL: https://leetcode.ca/2023-09-21-2863-Maximum-Length-of-Semi-Decreasing-Subarrays/ Title: Maximum Length of Semi-Decreasing Subarrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 3148 URL: https://leetcode.ca/2023-09-15-2852-Sum-of-Remoteness-of-All-Cells/ Title: Sum of Remoteness of All Cells Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 3111 URL: https://leetcode.ca/2023-09-09-2847-Smallest-Number-With-Given-Digit-Product/ Title: Smallest Number With Given Digit Product Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2890 URL: https://leetcode.ca/2023-09-27-2743-Count-Substrings-Without-Repeating-Character/ Title: Count Substrings Without Repeating Character Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2880 URL: https://leetcode.ca/2023-09-21-2737-Find-the-Closest-Marked-Node/ Title: Find the Closest Marked Node Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2875 URL: https://leetcode.ca/2023-09-12-2728-Count-Houses-in-a-Circular-Street/ Title: Count Houses in a Circular Street Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2865 URL: https://leetcode.ca/2023-08-29-2714-Find-Shortest-Path-with-K-Hops/ Title: Find Shortest Path with K Hops Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2843 URL: https://leetcode.ca/2023-08-04-2689-Extract-Kth-Character-From-The-Rope-Tree/ Title: Extract Kth Character From The Rope Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2835 URL: https://leetcode.ca/2023-07-20-2674-Split-a-Circular-Linked-List/ Title: Split a Circular Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2823 URL: https://leetcode.ca/2023-07-10-2664-The-Knights-Tour/ Title: The Knight's Tour Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR

2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2815 URL: https://leetcode.ca/2023-07-01-2655-Find-Maximal-Uncovered-Ranges/ Title: Find Maximal Uncovered Ranges Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2738 URL: https://leetcode.ca/2023-06-14-2638-Count-the-Number-of-K-Free-Subsets/ Title: Count the Number of K-Free Subsets Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2729 URL: https://leetcode.ca/2023-08-17-2702-Minimum-Operations-to-Make-Numbers-Non-positive/ Title: Minimum Operations to Make Numbers Non-positive Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2719 URL: https://leetcode.ca/2023-05-20-2613-Beautiful-Pairs/ Title: Beautiful Pairs Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===

Problem Number: 2702 URL: https://leetcode.ca/2023-03-25-2557-Maximum-Number-of-Integers-to-Choose-From-a-Range-II/ Title: Maximum Number of Integers to Choose From a Range II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===

Problem Number: 2693 URL: https://leetcode.ca/2023-03-16-2548-Maximum-Price-to-Fill-a-Bag/ Title: Maximum Price to Fill a Bag Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===

Problem Number: 2688 URL: https://leetcode.ca/2023-04-27-2590-Design-a-

Todo-List/ Title: Design a Todo List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2683 URL: https://leetcode.ca/2023-03-07-2539-Count-the-Number-of-Good-Subsequences/ Title: Count the Number of Good Subsequences Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2674 URL: https://leetcode.ca/2023-05-06-2599-Make-the-Prefix-Sum-Non-negative/ Title: Make the Prefix Sum Non-negative Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the

number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 2672 URL: https://leetcode.ca/2023-03-01-2533-Number-of-Good-Binary-Strings/ Title: Number of Good Binary Strings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 2667 URL: https://leetcode.ca/2023-02-20-2524-Maximum-Frequency-Score-of-a-Subarray/ Title: Maximum Frequency Score of a Subarray Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 2666 URL: https://leetcode.ca/2023-05-11-2604-Minimum-Time-to-Eat-All-Grains/ Title: Minimum Time to Eat All Grains Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 2658 URL: https://leetcode.ca/2023-02-15-2519-Count-the-Number-of-K-Big-Indices/ Title: Count the Number of K-Big Indices Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2653 URL: https://leetcode.ca/2023-02-06-2510-Check-if-There-is-a-Path-With-Equal-Number-of-0's-And-1's/ Title: Check if There is a Path With Equal Number of 0's And 1's Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2644 URL: https://leetcode.ca/2023-02-01-2505-Bitwise-OR-of-All-Subsequence-Sums/ Title: Bitwise OR of All Subsequence Sums Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2638 URL: https://leetcode.ca/2023-01-22-2495-Number-of-Subarrays-Having-Even-Product/ Title: Number of Subarrays Having Even Product Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2629 URL: https://leetcode.ca/2023-01-16-2489-Number-of-Substrings-With-Fixed-Ratio/ Title: Number of Substrings With Fixed Ratio Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2623 URL: https://leetcode.ca/2023-01-06-2479-Maximum-XOR-of-Two-Non-Overlapping-Subtrees/ Title: Maximum XOR of Two Non-Overlapping Subtrees Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2617 URL: https://leetcode.ca/2023-03-02-2534-Time-Taken-to-Cross-the-Door/ Title: Time Taken to Cross the Door Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2612 URL: https://leetcode.ca/2022-12-31-2473-Minimum-Cost-to-Buy-Apples/ Title: Minimum Cost to Buy Apples Problem Description:

You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2607 URL: https://leetcode.ca/2022-12-22-2464-Minimum-Subarrays-in-a-Valid-Split/ Title: Minimum Subarrays in a Valid Split Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2593 URL: https://leetcode.ca/2022-12-08-2450-Number-of-Distinct-Binary-Strings-After-Applying-Operations/ Title: Number of Distinct Binary Strings After Applying Operations Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where

XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2584 URL: https://leetcode.ca/2022-12-03-2445-Number-of-Nodes-With-Value-One/ Title: Number of Nodes With Value One Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2579 URL: https://leetcode.ca/2022-11-24-2436-Minimum-Split-Into-Subarrays-With-GCD-Greater-Than-One/ Title: Minimum Split Into Subarrays With GCD Greater Than One Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2570 URL: https://leetcode.ca/2022-11-19-2431-Maximize-Total-Tastiness-of-Purchased-Fruits/ Title: Maximize Total Tastiness of Purchased Fruits Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2565 URL: https://leetcode.ca/2022-11-10-2422-Merge-Operations-to-Turn-Array-Into-a-Palindrome/ Title: Merge Operations to Turn Array Into a Palindrome Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2560 URL: https://leetcode.ca/2022-11-05-2417-Closest-Fair-Integer/ Title: Closest Fair Integer Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2555 URL: https://leetcode.ca/2022-10-27-2408-Design-SQL/ Title: Design SQL Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 2540 URL: https://leetcode.ca/2023-06-23-2647-Color-the-Triangle-Red/ Title: Color the Triangle Red Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 2537 URL: https://leetcode.ca/2022-10-22-2403-Minimum-Time-to-Kill-All-Monsters/ Title: Minimum Time to Kill All Monsters Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 2535 URL: https://leetcode.ca/2022-10-12-2393-Count-

Strictly-Increasing-Subarrays/ Title: Count Strictly Increasing Subarrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2522 URL: https://leetcode.ca/2022-10-06-2387-Median-of-a-Row-Wise-Sorted-Matrix/ Title: Median of a Row Wise Sorted Matrix Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2517 URL: https://leetcode.ca/2022-09-27-2378-Choose-Edges-to-Maximize-Score-in-a-Tree/ Title: Choose Edges to Maximize Score in a Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2506 URL: https://leetcode.ca/2022-09-20-2371-Minimize-Maximum-Value-in-a-Grid/ Title: Minimize Maximum Value in a Grid Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2500 URL: https://leetcode.ca/2022-09-10-2361-Minimum-Costs-Using-the-Train-Line/ Title: Minimum Costs Using the Train Line Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2490 URL: https://leetcode.ca/2022-09-04-2355-Maximum-Number-of-Books-You-Can-Take/ Title: Maximum Number of Books You Can Take Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2489 URL: https://leetcode.ca/2022-12-17-2459-Sort-Array-by-Moving-Items-to-Empty-Space/ Title: Sort Array by Moving Items to Empty Space Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2485 URL: https://leetcode.ca/2022-08-25-2345-Finding-the-Number-of-Visible-Mountains/ Title: Finding the Number of Visible Mountains Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2474 URL: https://leetcode.ca/2022-08-20-2340-Minimum-Adjacent-Swaps-to-Make-a-Valid-Array/ Title: Minimum Adjacent Swaps to Make a Valid Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2468 URL: https://leetcode.ca/2022-08-10-2330-Valid-Palindrome-IV/ Title: Valid Palindrome IV Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2458 URL: https://leetcode.ca/2022-08-03-2323-Find-Minimum-Time-to-Finish-All-Jobs-II/ Title: Find Minimum Time to Finish All Jobs II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2426 URL: https://leetcode.ca/2022-07-02-2291-Maximum-Profit-From-Trading-Stocks/ Title: Maximum Profit From Trading Stocks Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2425 URL: https://leetcode.ca/2022-06-23-2282-Number-of-People-That-Can-Be-Seen-in-a-Grid/ Title: Number of People That Can Be Seen in a Grid Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2420 URL: https://leetcode.ca/2022-06-18-2277-Closest-Node-to-Path-in-Tree/ Title: Closest Node to Path in Tree Problem Description:

You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2405 URL: https://leetcode.ca/2022-06-09-2268-Minimum-Number-of-Keypresses/ Title: Minimum Number of Keypresses Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2399 URL: https://leetcode.ca/2022-07-24-2313-Minimum-Flips-in-Binary-Tree-to-Get-Result/ Title: Minimum Flips in Binary Tree to Get Result Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2396 URL: https://leetcode.ca/2022-05-26-2254-Design-Video-Sharing-Platform/ Title: Design Video Sharing Platform Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2385 URL: https://leetcode.ca/2022-05-09-2237-Count-Positions-on-Street-With-Required-Brightness/ Title: Count Positions on Street With Required Brightness Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2369 URL: https://leetcode.ca/2022-04-21-2219-Maximum-Sum-Score-of-Array/ Title: Maximum Sum Score of Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2354 URL: https://leetcode.ca/2022-04-16-2214-Minimum-Health-to-Beat-Game/ Title: Minimum Health to Beat Game Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2322 URL: https://leetcode.ca/2022-03-17-2184-Number-of-Ways-to-Build-Sturdy-Brick-Wall/ Title: Number of Ways to Build Sturdy Brick Wall Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2314 URL: https://leetcode.ca/2022-03-07-2174-Remove-All-Ones-With-Row-and-Column-Flips-II/ Title: Remove All Ones With Row and Column Flips II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2303 URL: https://leetcode.ca/2022-03-01-2168-Unique-Substrings-With-Equal-Digit-Frequency/ Title: Unique Substrings With Equal Digit Frequency Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2297 URL: https://leetcode.ca/2022-02-19-2158-Amount-of-New-Area-Painted-Each-Day/ Title: Amount of New Area Painted Each Day Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2287 URL: https://leetcode.ca/2022-02-13-2152-Minimum-Number-of-Lines-to-Cover-Points/ Title: Minimum Number of Lines to Cover Points Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2282 URL: https://leetcode.ca/2022-02-04-2143-Choose-Numbers-From-Two-Arrays-in-Range/ Title: Choose Numbers From Two Arrays in Range Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2273 URL: https://leetcode.ca/2022-01-29-2137-Pour-Water-Between-Buckets-to-Make-Water-Levels-Equal/ Title: Pour Water Between Buckets to Make Water Levels Equal Problem Description: You are given a 2D

integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2268 URL: https://leetcode.ca/2022-01-20-2128-Remove-All-Ones-With-Row-and-Column-Flips/ Title: Remove All Ones With Row and Column Flips Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2259 URL: https://leetcode.ca/2022-01-15-2123-Minimum-Operations-to-Remove-Adjacent-Ones-in-Matrix/ Title: Minimum Operations to Remove Adjacent Ones in Matrix Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR

is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 2258 URL: https://leetcode.ca/2022-01-05-2113-Elements-in-Array-After-Removing-and-Replacing-Elements/ Title: Elements in Array After Removing and Replacing Elements Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 2247 URL: https://leetcode.ca/2021-12-30-2107-Number-of-Unique-Flavors-After-Sharing-K-Candies/ Title: Number of Unique Flavors After Sharing K Candies Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2242 URL: https://leetcode.ca/2021-12-21-2098-Subsequence-of-Size-K-With-the-Largest-Even-Sum/ Title: Subsequence of Size K With the Largest Even Sum Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2230 URL: https://leetcode.ca/2021-12-16-2093-Minimum-Cost-to-Reach-City-With-Discounts/ Title: Minimum Cost to Reach City With Discounts Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2225 URL: https://leetcode.ca/2021-12-06-2083-Substrings-That-Begin-and-End-With-the-Same-Letter/ Title: Substrings That Begin and End With the Same Letter Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2218 URL: https://leetcode.ca/2021-11-30-2077-Paths-in-Maze-That-Lead-to-Same-Room/ Title: Paths in Maze That Lead to Same Room Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2209 URL: https://leetcode.ca/2021-11-20-2067-Number-of-Equal-Count-Substrings/ Title: Number of Equal Count Substrings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2203 URL: https://leetcode.ca/2021-11-14-2061-Number-of-Spaces-Cleaning-Robot-Cleaned/ Title: Number of Spaces Cleaning Robot Cleaned Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2153 URL: https://leetcode.ca/2021-09-19-2005-Subtree-Removal-Game-with-Fibonacci-Tree/ Title: Subtree Removal Game with Fibonacci Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2142 URL: https://leetcode.ca/2021-09-29-2015-Average-Height-of-Buildings-in-Each-Segment/ Title: Average Height of Buildings in Each Segment Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2141 URL: https://leetcode.ca/2021-09-13-1999-Smallest-Greater-Multiple-Made-of-Two-Digits/ Title: Smallest Greater Multiple Made

of Two Digits Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2126 URL: https://leetcode.ca/2021-08-18-1973-Count-Nodes-Equal-to-Sum-of-Descendants/ Title: Count Nodes Equal to Sum of Descendants Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2111 URL: https://leetcode.ca/2021-08-16-1966-Binary-Searchable-Numbers-in-an-Unsorted-Array/ Title: Binary Searchable Numbers in an Unsorted Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise

XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2082 URL: https://leetcode.ca/2021-11-05-2052-Minimum-Cost-to-Separate-Sentence-Into-Rows/ Title: Minimum Cost to Separate Sentence Into Rows Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2075 URL: https://leetcode.ca/2021-10-05-2021-Brightest-Position-on-Street/ Title: Brightest Position on Street Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2074 URL: https://leetcode.ca/2021-08-03-1924-Erect-the-Fence-II/ Title: Erect the Fence II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2073 URL: https://leetcode.ca/2021-08-13-1956-Minimum-Time-For-K-Virus-Variants-to-Spread/ Title: Minimum Time For K Virus Variants to Spread Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2072 URL: https://leetcode.ca/2021-08-11-1950-Maximum-of-Minimum-Values-in-All-Subarrays/ Title: Maximum of Minimum Values in All Subarrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2071 URL: https://leetcode.ca/2021-08-08-1940-Longest-Common-Subsequence-Between-Sorted-Arrays/ Title: Longest Common Subsequence Between Sorted Arrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2070 URL: https://leetcode.ca/2021-08-06-1933-Check-if-String-Is-Decomposable-Into-Value-Equal-Substrings/ Title: Check if String Is Decomposable Into Value-Equal Substrings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2069 URL: https://leetcode.ca/2021-08-01-1918-Kth-Smallest-Subarray-Sum/ Title: Kth Smallest Subarray Sum Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2065 URL: https://leetcode.ca/2022-07-18-2307-Check-for-Contradictions-in-Equations/ Title: Check for Contradictions in Equations Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2062 URL: https://leetcode.ca/2021-07-29-1908-Game-of-Nim/ Title: Game of Nim Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2052 URL: https://leetcode.ca/2021-07-28-1902-Depth-of-BST-Given-Insertion-Order/ Title: Depth of BST Given Insertion Order Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in

a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2045 URL: https://leetcode.ca/2021-07-24-1891-Cutting-Ribbons/ Title: Cutting Ribbons Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2036 URL: https://leetcode.ca/2021-07-23-1885-Count-Pairs-in-Two-Arrays/ Title: Count Pairs in Two Arrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2029 URL: https://leetcode.ca/2021-07-19-1874-Minimize-Product-Sum-of-Two-Arrays/ Title: Minimize Product Sum of Two Arrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2019 URL: https://leetcode.ca/2021-07-17-1868-Product-of-Two-Run-Length-Encoded-Arrays/ Title: Product of Two Run-Length Encoded Arrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2009 URL: https://leetcode.ca/2021-07-14-1858-Longest-Word-With-All-Prefixes/ Title: Longest Word With All Prefixes Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 2007 URL: https://leetcode.ca/2022-05-19-2247-Maximum-Cost-of-Trip-With-K-Highways/ Title: Maximum Cost of Trip With K Highways Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 2003 URL: https://leetcode.ca/2021-07-13-1852-Distinct-Numbers-in-Each-Subarray/ Title: Distinct Numbers in Each Subarray Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1997 URL: https://leetcode.ca/2021-07-03-1842-Next-Palindrome-Using-Same-Digits/ Title: Next Palindrome Using Same Digits Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1992 URL: https://leetcode.ca/2021-10-30-2046-Sort-Linked-List-Already-Sorted-Using-Absolute-Values/ Title: Sort Linked List Already Sorted Using Absolute Values Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1982 URL: https://leetcode.ca/2021-06-27-1836-Remove-Duplicates-From-an-Unsorted-Linked-List/ Title: Remove Duplicates From an Unsorted Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1980 URL: https://leetcode.ca/2021-06-17-1826-Faulty-Sensor/ Title: Faulty Sensor Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1979 URL: https://leetcode.ca/2021-09-03-1989-Maximum-Number-of-People-That-Can-Be-Caught-in-Tag/ Title: Maximum Number of People That Can Be Caught in Tag Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1969 URL: https://leetcode.ca/2021-06-11-1820-Maximum-Number-of-Accepted-Invitations/ Title: Maximum Number of Accepted Invitations Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1959 URL: https://leetcode.ca/2021-06-01-1810-Minimum-Path-Cost-in-a-Hidden-Grid/ Title: Minimum Path Cost in a Hidden Grid Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1949 URL: https://leetcode.ca/2021-05-26-1804-Implement-Trie-II-(Prefix-Tree)/ Title: Implement Trie II (Prefix Tree) Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1943 URL: https://leetcode.ca/2021-05-16-1794-Count-Pairs-of-Equal-Substrings-With-Minimum-Difference/ Title: Count Pairs of Equal Substrings With Minimum Difference Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1937 URL: https://leetcode.ca/2021-05-10-1788-Maximize-the-Beauty-of-the-Garden/ Title: Maximize the Beauty of the Garden Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1931 URL: https://leetcode.ca/2021-04-30-1778-Shortest-Path-in-a-Hidden-Grid/ Title: Shortest Path in a Hidden Grid Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1919 URL: https://leetcode.ca/2021-04-24-1772-Sort-Features-by-Popularity/ Title: Sort Features by Popularity Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1909 URL: https://leetcode.ca/2021-04-14-1762-Buildings-With-an-Ocean-View/ Title: Buildings With an Ocean View Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1893 URL: https://leetcode.ca/2021-03-29-1746-Maximum-Subarray-Sum-After-One-Operation/ Title: Maximum Subarray Sum After One Operation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1883 URL: https://leetcode.ca/2021-03-23-1740-Find-Distance-in-a-Binary-Tree/ Title: Find Distance in a Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1865 URL: https://leetcode.ca/2021-03-07-1724-Checking-Existence-of-Edge-Length-Limited-Paths-II/ Title: Checking Existence of Edge Length Limited Paths II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1847 URL: https://leetcode.ca/2021-02-11-1708-Largest-Subarray-Length-K/ Title: Largest Subarray Length K Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1838 URL: https://leetcode.ca/2020-07-24-1698-Number-of-Distinct-Substrings-in-a-String/ Title: Number of Distinct Substrings in a String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1828 URL: https://leetcode.ca/2020-07-18-1692-Count-Ways-to-Distribute-Candies/ Title: Count Ways to Distribute Candies Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1822 URL: https://leetcode.ca/2020-07-08-1682-Longest-Palindromic-Subsequence-II/ Title: Longest Palindromic Subsequence II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1816 URL: https://leetcode.ca/2020-07-02-1676-Lowest-Common-Ancestor-of-a-Binary-Tree-IV/ Title: Lowest Common Ancestor of a Binary Tree IV Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1810 URL: https://leetcode.ca/2020-06-22-1666-Change-the-Root-of-a-Binary-Tree/ Title: Change the Root of a Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1809 URL: https://leetcode.ca/2021-02-24-1714-Sum-Of-Special-Evenly-Spaced-Elements-In-Array/ Title: Sum Of Special Evenly-Spaced Elements In Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1796 URL: https://leetcode.ca/2020-06-16-1660-Correct-a-Binary-Tree/ Title: Correct a Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1790 URL: https://leetcode.ca/2020-06-06-1650-Lowest-Common-Ancestor-of-a-Binary-Tree-III/ Title: Lowest Common Ancestor of a Binary Tree III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1780 URL: https://leetcode.ca/2020-05-31-1644-Lowest-Common-Ancestor-of-a-Binary-Tree-II/ Title: Lowest Common Ancestor of a Binary Tree II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1774 URL: https://leetcode.ca/2020-05-21-1634-Add-Two-Polynomials-Represented-as-Linked-Lists/ Title: Add Two Polynomials Represented as Linked Lists Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1768 URL: https://leetcode.ca/2020-05-15-1628-Design-an-Expression-Tree-With-Evaluate-Function/ Title: Design an Expression Tree With Evaluate Function Problem Description: You are given a 2D integer array

coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1750 URL: https://leetcode.ca/2020-04-29-1612-Check-If-Two-Expression-Trees-are-Equivalent/ Title: Check If Two Expression Trees are Equivalent Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1745 URL: https://leetcode.ca/2020-04-19-1602-Find-Nearest-Right-Node-in-Binary-Tree/ Title: Find Nearest Right Node in Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation.

Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1736 URL: https://leetcode.ca/2020-04-14-1597-Build-Binary-Expression-Tree-From-Infix-Expression/ Title: Build Binary Expression Tree From Infix Expression Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1729 URL: https://leetcode.ca/2020-04-03-1586-Binary-Search-Tree-Iterator-II/ Title: Binary Search Tree Iterator II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1703 URL: https://leetcode.ca/2020-03-12-1564-Put-Boxes-Into-the-Warehouse-I/ Title: Put Boxes Into the Warehouse I Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1687 URL: https://leetcode.ca/2020-02-25-1548-The-Most-Similar-Path-in-a-Graph/ Title: The Most Similar Path in a Graph Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1681 URL: https://leetcode.ca/2020-02-15-1538-Guess-the-Majority-in-a-Hidden-Array/ Title: Guess the Majority in a Hidden Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1672 URL: https://leetcode.ca/2020-02-10-1533-Find-the-Index-of-the-Large-Integer/ Title: Find the Index of the Large Integer Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1655 URL: https://leetcode.ca/2020-01-24-1516-Move-Sub-Tree-of-N-Ary-Tree/ Title: Move Sub-Tree of N-Ary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1650 URL: https://leetcode.ca/2020-01-14-1506-Find-Root-of-N-Ary-Tree/ Title: Find Root of N-Ary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1640 URL: https://leetcode.ca/2020-01-08-1500-Design-a-File-Sharing-System/ Title: Design a File Sharing System Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1634 URL: https://leetcode.ca/2019-12-29-1490-Clone-N-ary-Tree/ Title: Clone N-ary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1624 URL: https://leetcode.ca/2019-12-24-1485-Clone-Binary-Tree-With-Random-Pointer/ Title: Clone Binary Tree With Random Pointer Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1618 URL: https://leetcode.ca/2019-12-13-1474-Delete-N-Nodes-After-M-Nodes-of-a-Linked-List/ Title: Delete N Nodes After M Nodes of a Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1432 URL: https://leetcode.ca/2019-10-30-1430-Check-If-a-String-Is-a-Valid-Sequence-from-Root-to-Leaves-Path-in-a-Binary-Tree/ Title: Check If a String Is a Valid Sequence from Root to Leaves Path in a Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation.

Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1419 URL: https://leetcode.ca/2022-06-04-2263-Make-Array-Non-decreasing-or-Non-increasing/ Title: Make Array Non-decreasing or Non-increasing Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1416 URL: https://leetcode.ca/2022-05-01-2229-Check-if-an-Array-Is-Consecutive/ Title: Check if an Array Is Consecutive Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1404 URL: https://leetcode.ca/2019-05-18-1265-Print-Immutable-Linked-List-in-Reverse/ Title: Print Immutable Linked List in Reverse Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1385 URL: https://leetcode.ca/2022-03-22-2189-Number-of-Ways-to-Build-House-of-Cards/ Title: Number of Ways to Build House of Cards Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1384 URL: https://leetcode.ca/2020-05-05-1618-Maximum-Font-to-Fit-a-Sentence-in-a-Screen/ Title: Maximum Font to Fit a Sentence in a Screen Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1383 URL: https://leetcode.ca/2022-03-31-2198-Number-of-Single-Divisor-Triplets/ Title: Number of Single Divisor Triplets Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1374 URL: https://leetcode.ca/2019-10-28-1428-Leftmost-Column-with-at-Least-a-One/ Title: Leftmost Column with at Least a One Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1366 URL: https://leetcode.ca/2019-10-29-1429-First-Unique-Number/ Title: First Unique Number Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1347 URL: https://leetcode.ca/2022-04-06-2204-Distance-to-a-Cycle-in-Undirected-Graph/ Title: Distance to a Cycle in Undirected Graph Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1280 URL: https://leetcode.ca/2019-02-18-1176-Diet-Plan-Performance/ Title: Diet Plan Performance Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1271 URL: https://leetcode.ca/2019-04-19-1236-Web-Crawler/ Title: Web Crawler Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1233 URL: https://leetcode.ca/2019-05-27-1274-Number-of-Ships-in-a-Rectangle/ Title: Number of Ships in a Rectangle Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1213 URL: https://leetcode.ca/2019-05-12-1259-Handshakes-That-Don't-Cross/ Title: Handshakes That Don't Cross Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1202 URL: https://leetcode.ca/2019-04-29-1246-Palindrome-Removal/ Title: Palindrome Removal Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1201 URL: https://leetcode.ca/2019-05-26-1273-Delete-Tree-Nodes/ Title: Delete Tree Nodes Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1200 URL: https://leetcode.ca/2019-05-25-1272-Remove-Interval/ Title: Remove Interval Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1191 URL: https://leetcode.ca/2019-05-11-1258-Synonymous-Sentences/ Title: Synonymous Sentences Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 1178 URL: https://leetcode.ca/2019-03-30-1216-Valid-Palindrome-III/ Title: Valid Palindrome III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 1177 URL: https://leetcode.ca/2019-04-28-1245-Tree-Diameter/ Title: Tree Diameter Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 1176 URL: https://leetcode.ca/2019-04-27-1244-Design-A-Leaderboard/ Title: Design A Leaderboard Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the

distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1167 URL: https://leetcode.ca/2019-03-13-1199-Minimum-Time-to-Build-Blocks/ Title: Minimum Time to Build Blocks Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1166 URL: https://leetcode.ca/2019-04-13-1230-Toss-Strange-Coins/ Title: Toss Strange Coins Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1152 URL: https://leetcode.ca/2019-02-25-1183-Maximum-Number-of-Ones/ Title: Maximum Number of Ones Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1151 URL: https://leetcode.ca/2019-03-29-1215-Stepping-Numbers/ Title: Stepping Numbers Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1150 URL: https://leetcode.ca/2019-03-28-1214-Two-Sum-BSTs/ Title: Two Sum BSTs Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1144 URL: https://leetcode.ca/2019-02-10-1168-Optimize-Water-Distribution-in-a-Village/ Title: Optimize Water Distribution in a Village Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1142 URL: https://leetcode.ca/2019-03-11-1197-Minimum-Knight-Moves/ Title: Minimum Knight Moves Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1124 URL: https://leetcode.ca/2019-01-26-1153-String-Transforms-Into-Another-String/ Title: String Transforms Into Another String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1118 URL: https://leetcode.ca/2018-12-25-1121-Divide-

Array-Into-Increasing-Sequences/ Title: Divide Array Into Increasing Sequences Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1102 URL: https://leetcode.ca/2019-01-23-1150-Check-If-a-Number-Is-Majority-Element-in-a-Sorted-Array/ Title: Check If a Number Is Majority Element in a Sorted Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1099 URL: https://leetcode.ca/2018-12-06-1102-Path-With-Maximum-Minimum-Value/ Title: Path With Maximum Minimum Value Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2)

as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1091 URL: https://leetcode.ca/2018-12-24-1120-Maximum-Average-Subtree/ Title: Maximum Average Subtree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1090 URL: https://leetcode.ca/2019-01-07-1134-Armstrong-Number/ Title: Armstrong Number Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1089 URL: https://leetcode.ca/2018-12-23-1119-Remove-Vowels-from-a-String/ Title: Remove Vowels from a String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1088 URL: https://leetcode.ca/2018-12-22-1118-Number-of-Days-in-a-Month/ Title: Number of Days in a Month Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1084 URL: https://leetcode.ca/2018-12-04-1100-Find-K-Length-Substrings-With-No-Repeated-Characters/ Title: Find K-Length Substrings With No Repeated Characters Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 1083 URL: https://leetcode.ca/2018-12-03-1099-Two-Sum-Less-Than-K/ Title: Two Sum Less Than K Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1077 URL: https://leetcode.ca/2018-11-22-1088-Confusing-Number-II/ Title: Confusing Number II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1076 URL: https://leetcode.ca/2018-11-21-1087-Brace-Expansion/ Title: Brace Expansion Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1074 URL: https://leetcode.ca/2018-11-20-1086-High-Five/ Title: High Five Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1069 URL: https://leetcode.ca/2018-10-21-1056-Confusing-Number/ Title: Confusing Number Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 1068 URL: https://leetcode.ca/2018-11-01-1067-Digit-Count-in-Range/ Title: Digit Count in Range Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR

is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1061 URL: https://leetcode.ca/2018-10-28-1063-Number-of-Valid-Subarrays/ Title: Number of Valid Subarrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 1052 URL: https://leetcode.ca/2018-10-22-1057-Campus-Bikes/ Title: Campus Bikes Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR

2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 865 URL: https://leetcode.ca/2017-04-02-489-Robot-Room-Cleaner/ Title: Robot Room Cleaner Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 850 URL: https://leetcode.ca/2017-11-07-708-Insert-into-a-Sorted-Circular-Linked-List/ Title: Insert into a Sorted Circular Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 818 URL: https://leetcode.ca/2018-02-07-800-Similar-RGB-Color/ Title: Similar RGB Color Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 791 URL: https://leetcode.ca/2018-01-14-776-Split-BST/ Title: Split BST Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 788 URL: https://leetcode.ca/2018-01-12-774-Minimize-Max-Distance-to-Gas-Station/ Title: Minimize Max Distance to Gas Station Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 786 URL: https://leetcode.ca/2017-11-01-702-Search-in-a-Sorted-Array-of-Unknown-Size/ Title: Search in a Sorted Array of Unknown Size Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 785 URL: https://leetcode.ca/2018-01-10-772-Basic-Calculator-III/ Title: Basic Calculator III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi,

yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 771 URL: https://leetcode.ca/2017-02-03-431-Encode-N-ary-Tree-to-Binary-Tree/ Title: Encode N-ary Tree to Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 765 URL: https://leetcode.ca/2017-01-31-428-Serialize-and-Deserialize-N-ary-Tree/ Title: Serialize and Deserialize N-ary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 762 URL: https://leetcode.ca/2017-12-29-760-Find-Anagram-Mappings/ Title: Find Anagram Mappings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 761 URL: https://leetcode.ca/2017-12-28-759-Employee-Free-Time/ Title: Employee Free Time Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 760 URL: https://leetcode.ca/2017-12-27-758-Bold-Words-in-String/ Title: Bold Words in String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 758 URL: https://leetcode.ca/2017-01-29-426-Convert-Binary-Search-Tree-to-Sorted-Doubly-Linked-List/ Title: Convert Binary Search Tree to Sorted Doubly Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 756 URL: https://leetcode.ca/2017-12-24-755-Pour-Water/ Title: Pour Water Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 752 URL: https://leetcode.ca/2017-12-20-751-IP-to-CIDR/ Title: IP to CIDR Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 751 URL: https://leetcode.ca/2017-12-19-750-Number-Of-Corner-Rectangles/ Title: Number Of Corner Rectangles Problem Description:

You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 743 URL: https://leetcode.ca/2017-12-11-742-Closest-Leaf-in-a-Binary-Tree/ Title: Closest Leaf in a Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 737 URL: https://leetcode.ca/2017-12-06-737-Sentence-Similarity-II/ Title: Sentence Similarity II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where

XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 734 URL: https://leetcode.ca/2017-12-03-734-Sentence-Similarity/ Title: Sentence Similarity Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 727 URL: https://leetcode.ca/2017-11-26-727-Minimum-Window-Subsequence/ Title: Minimum Window Subsequence Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR

2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 723 URL: https://leetcode.ca/2017-11-22-723-Candy-Crush/ Title: Candy Crush Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 716 URL: https://leetcode.ca/2017-11-15-716-Max-Stack/ Title: Max Stack Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 711 URL: https://leetcode.ca/2017-11-10-711-Number-of-Distinct-Islands-II/ Title: Number of Distinct Islands II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 694 URL: https://leetcode.ca/2017-10-24-694-Number-of-Distinct-Islands/ Title: Number of Distinct Islands Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 683 URL: https://leetcode.ca/2017-10-13-683-K-Empty-Slots/ Title: K Empty Slots Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 681 URL: https://leetcode.ca/2017-10-11-681-Next-Closest-Time/ Title: Next Closest Time Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 666 URL: https://leetcode.ca/2017-09-26-666-Path-Sum-IV/ Title: Path Sum IV Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise

XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 663 URL: https://leetcode.ca/2017-09-23-663-Equal-Tree-Partition/ Title: Equal Tree Partition Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 660 URL: https://leetcode.ca/2017-09-20-660-Remove-9/ Title: Remove 9 Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 656 URL: https://leetcode.ca/2017-09-16-656-Coin-Path/ Title: Coin Path Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 651 URL: https://leetcode.ca/2017-09-11-651-4-Keys-Keyboard/ Title: 4 Keys Keyboard Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 644 URL: https://leetcode.ca/2017-09-04-644-Maximum-Average-Subarray-II/ Title: Maximum Average Subarray II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 642 URL: https://leetcode.ca/2017-09-02-642-Design-Search-Autocomplete-System/ Title: Design Search Autocomplete System Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 635 URL: https://leetcode.ca/2017-08-26-635-Design-Log-

Storage-System/ Title: Design Log Storage System Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 634 URL: https://leetcode.ca/2017-08-25-634-Find-the-Derangement-of-An-Array/ Title: Find the Derangement of An Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 631 URL: https://leetcode.ca/2017-08-22-631-Design-Excel-Sum-Formula/ Title: Design Excel Sum Formula Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 625 URL: https://leetcode.ca/2017-08-16-625-Minimum-Factorization/ Title: Minimum Factorization Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 624 URL: https://leetcode.ca/2017-08-15-624-Maximum-Distance-in-Arrays/ Title: Maximum Distance in Arrays Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 616 URL: https://leetcode.ca/2017-08-07-616-Add-Bold-Tag-in-String/ Title: Add Bold Tag in String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 588 URL: https://leetcode.ca/2017-07-10-588-Design-In-Memory-File-System/ Title: Design In-Memory File System Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 582 URL: https://leetcode.ca/2017-07-04-582-Kill-Process/ Title: Kill Process Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 573 URL: https://leetcode.ca/2017-06-25-573-Squirrel-Simulation/ Title: Squirrel Simulation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 568 URL: https://leetcode.ca/2017-06-20-568-Maximum-Vacation-Days/ Title: Maximum Vacation Days Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 562 URL: https://leetcode.ca/2017-06-14-562-Longest-Line-of-Consecutive-One-in-Matrix/ Title: Longest Line of Consecutive One in Matrix Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 555 URL: https://leetcode.ca/2017-06-07-555-Split-

Concatenated-Strings/ Title: Split Concatenated Strings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 550 URL: https://leetcode.ca/2021-03-13-1730-Shortest-Path-to-Get-Food/ Title: Shortest Path to Get Food Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 549 URL: https://leetcode.ca/2017-06-01-549-Binary-Tree-Longest-Consecutive-Sequence-II/ Title: Binary Tree Longest Consecutive Sequence II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 548 URL: https://leetcode.ca/2017-05-31-548-Split-Array-with-Equal-Sum/ Title: Split Array with Equal Sum Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 545 URL: https://leetcode.ca/2017-05-28-545-Boundary-of-Binary-Tree/ Title: Boundary of Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 544 URL: https://leetcode.ca/2017-05-27-544-Output-Contest-Matches/ Title: Output Contest Matches Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 536 URL: https://leetcode.ca/2017-05-19-536-Construct-Binary-Tree-from-String/ Title: Construct Binary Tree from String Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 533 URL: https://leetcode.ca/2017-05-16-533-Lonely-Pixel-II/ Title: Lonely Pixel II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 531 URL: https://leetcode.ca/2017-05-14-531-Lonely-Pixel-I/ Title: Lonely Pixel I Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 527 URL: https://leetcode.ca/2017-05-10-527-Word-Abbreviation/ Title: Word Abbreviation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 519 URL: https://leetcode.ca/2021-08-28-1983-Widest-Pair-of-Indices-With-Equal-Range-Sum/ Title: Widest Pair of Indices With Equal Range Sum Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 512 URL: https://leetcode.ca/2021-10-20-2036-Maximum-Alternating-Subarray-Sum/ Title: Maximum Alternating Subarray Sum

Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 511 URL: https://leetcode.ca/2018-10-24-1059-All-Paths-from-Source-Lead-to-Destination/ Title: All Paths from Source Lead to Destination Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 510 URL: https://leetcode.ca/2021-10-15-2031-Count-Subarrays-With-More-Ones-Than-Zeros/ Title: Count Subarrays With More Ones Than Zeros Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise

XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 505 URL: https://leetcode.ca/2017-04-18-505-The-Maze-II/ Title: The Maze II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 499 URL: https://leetcode.ca/2017-04-12-499-The-Maze-III/ Title: The Maze III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 490 URL: https://leetcode.ca/2017-04-03-490-The-Maze/ Title: The Maze Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 487 URL: https://leetcode.ca/2017-03-31-487-Max-Consecutive-Ones-II/ Title: Max Consecutive Ones II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 484 URL: https://leetcode.ca/2017-03-28-484-Find-Permutation/ Title: Find Permutation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 471 URL: https://leetcode.ca/2017-03-15-471-Encode-String-with-Shortest-Length/ Title: Encode String with Shortest Length Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 469 URL: https://leetcode.ca/2017-03-13-469-Convex-

Polygon/ Title: Convex Polygon Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 465 URL: https://leetcode.ca/2017-03-09-465-Optimal-Account-Balancing/ Title: Optimal Account Balancing Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 444 URL: https://leetcode.ca/2017-02-16-444-Sequence-Reconstruction/ Title: Sequence Reconstruction Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 439 URL: https://leetcode.ca/2017-02-11-439-Ternary-Expression-Parser/ Title: Ternary Expression Parser Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 425 URL: https://leetcode.ca/2017-01-28-425-Word-Squares/ Title: Word Squares Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR

2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 422 URL: https://leetcode.ca/2017-01-25-422-Valid-Word-Square/ Title: Valid Word Square Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 418 URL: https://leetcode.ca/2017-01-21-418-Sentence-Screen-Fitting/ Title: Sentence Screen Fitting Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 411 URL: https://leetcode.ca/2017-01-14-411-Minimum-Unique-Word-Abbreviation/ Title: Minimum Unique Word Abbreviation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 408 URL: https://leetcode.ca/2017-01-11-408-Valid-Word-Abbreviation/ Title: Valid Word Abbreviation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 370 URL: https://leetcode.ca/2016-12-04-370-Range-Addition/ Title: Range Addition Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 369 URL: https://leetcode.ca/2016-12-03-369-Plus-One-Linked-List/ Title: Plus One Linked List Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 366 URL: https://leetcode.ca/2016-11-30-366-Find-Leaves-of-Binary-Tree/ Title: Find Leaves of Binary Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR

y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 364 URL: https://leetcode.ca/2016-11-28-364-Nested-List-Weight-Sum-II/ Title: Nested List Weight Sum II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 361 URL: https://leetcode.ca/2016-11-25-361-Bomb-Enemy/ Title: Bomb Enemy Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 360 URL: https://leetcode.ca/2016-11-24-360-Sort-Transformed-Array/ Title: Sort Transformed Array Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 358 URL: https://leetcode.ca/2016-11-22-358-Rearrange-String-k-Distance-Apart/ Title: Rearrange String k Distance Apart Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 356 URL: https://leetcode.ca/2016-11-20-356-Line-Reflection/ Title: Line Reflection Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 351 URL: https://leetcode.ca/2016-11-15-351-Android-Unlock-Patterns/ Title: Android Unlock Patterns Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 348 URL: https://leetcode.ca/2016-11-12-348-Design-Tic-Tac-Toe/ Title: Design Tic-Tac-Toe Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 346 URL: https://leetcode.ca/2016-11-10-346-Moving-Average-from-Data-Stream/ Title: Moving Average from Data Stream Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 340 URL: https://leetcode.ca/2016-11-04-340-Longest-

Substring-with-At-Most-K-Distinct-Characters/ Title: Longest Substring with At Most K Distinct Characters Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 339 URL: https://leetcode.ca/2016-11-03-339-Nested-List-Weight-Sum/ Title: Nested List Weight Sum Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 333 URL: https://leetcode.ca/2016-10-28-333-Largest-BST-Subtree/ Title: Largest BST Subtree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR

is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 325 URL: https://leetcode.ca/2016-10-20-325-Maximum-Size-Subarray-Sum-Equals-k/ Title: Maximum Size Subarray Sum Equals k Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 320 URL: https://leetcode.ca/2016-10-15-320-Generalized-Abbreviation/ Title: Generalized Abbreviation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 317 URL: https://leetcode.ca/2016-10-12-317-Shortest-Distance-from-All-Buildings/ Title: Shortest Distance from All Buildings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 314 URL: https://leetcode.ca/2016-10-09-314-Binary-Tree-Vertical-Order-Traversal/ Title: Binary Tree Vertical Order Traversal Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 311 URL: https://leetcode.ca/2016-10-06-311-Sparse-Matrix-Multiplication/ Title: Sparse Matrix Multiplication Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 305 URL: https://leetcode.ca/2016-09-30-305-Number-of-Islands-II/ Title: Number of Islands II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 302 URL: https://leetcode.ca/2016-09-27-302-Smallest-Rectangle-Enclosing-Black-Pixels/ Title: Smallest Rectangle Enclosing Black Pixels Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 298 URL: https://leetcode.ca/2016-09-23-298-Binary-Tree-Longest-Consecutive-Sequence/ Title: Binary Tree Longest Consecutive Sequence Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 296 URL: https://leetcode.ca/2016-09-21-296-Best-Meeting-Point/ Title: Best Meeting Point Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 294 URL: https://leetcode.ca/2016-09-19-294-Flip-Game-II/ Title: Flip Game II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 293 URL: https://leetcode.ca/2016-09-18-293-Flip-Game/ Title: Flip Game Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise

XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 291 URL: https://leetcode.ca/2016-09-16-291-Word-Pattern-II/ Title: Word Pattern II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 288 URL: https://leetcode.ca/2016-09-13-288-Unique-Word-Abbreviation/ Title: Unique Word Abbreviation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR

2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 286 URL: https://leetcode.ca/2016-09-11-286-Walls-and-Gates/ Title: Walls and Gates Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 285 URL: https://leetcode.ca/2016-09-10-285-Inorder-Successor-in-BST/ Title: Inorder Successor in BST Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two

pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 281 URL: https://leetcode.ca/2016-09-06-281-Zigzag-Iterator/ Title: Zigzag Iterator Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 280 URL: https://leetcode.ca/2016-09-05-280-Wiggle-Sort/ Title: Wiggle Sort Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 277 URL: https://leetcode.ca/2016-09-02-277-Find-the-Celebrity/ Title: Find the Celebrity Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 276 URL: https://leetcode.ca/2016-09-01-276-Paint-Fence/ Title: Paint Fence Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

=== Problem Number: 272 URL: https://leetcode.ca/2016-08-28-272-Closest-Binary-Search-Tree-Value-II/ Title: Closest Binary Search Tree Value II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===

Problem Number: 271 URL: https://leetcode.ca/2016-08-27-271-Encode-and-Decode-Strings/ Title: Encode and Decode Strings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===

Problem Number: 270 URL: https://leetcode.ca/2016-08-26-270-Closest-Binary-Search-Tree-Value/ Title: Closest Binary Search Tree Value Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 269 URL: https://leetcode.ca/2016-08-25-269-Alien-Dictionary/ Title: Alien Dictionary Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 267 URL: https://leetcode.ca/2016-08-23-267-Palindrome-Permutation-II/ Title: Palindrome Permutation II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 266 URL: https://leetcode.ca/2016-08-22-266-Palindrome-Permutation/ Title: Palindrome Permutation Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 265 URL: https://leetcode.ca/2016-08-21-265-Paint-House-II/ Title: Paint House II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 261 URL: https://leetcode.ca/2016-08-17-261-Graph-Valid-Tree/ Title: Graph Valid Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 259 URL: https://leetcode.ca/2016-08-15-259-3Sum-Smaller/ Title: 3Sum Smaller Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

===
Problem Number: 256 URL: https://leetcode.ca/2016-08-12-256-Paint-House/ Title: Paint House Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates

of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 255 URL: https://leetcode.ca/2016-08-11-255-Verify-Preorder-Sequence-in-Binary-Search-Tree/ Title: Verify Preorder Sequence in Binary Search Tree Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

== Problem Number: 254 URL: https://leetcode.ca/2016-08-10-254-Factor-Combinations/ Title: Factor Combinations Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 253 URL: https://leetcode.ca/2016-08-09-253-Meeting-Rooms-II/ Title: Meeting Rooms II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 252 URL: https://leetcode.ca/2016-08-08-252-Meeting-Rooms/ Title: Meeting Rooms Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 251 URL: https://leetcode.ca/2016-08-07-251-Flatten-2D-Vector/ Title: Flatten 2D Vector Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 250 URL: https://leetcode.ca/2016-08-06-250-Count-Univalue-Subtrees/ Title: Count Univalue Subtrees Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 249 URL: https://leetcode.ca/2016-08-05-249-Group-Shifted-Strings/ Title: Group Shifted Strings Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 248 URL: https://leetcode.ca/2016-08-04-248-Strobogrammatic-Number-III/ Title: Strobogrammatic Number III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k.   Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 247 URL: https://leetcode.ca/2016-08-03-247-Strobogrammatic-Number-II/ Title: Strobogrammatic Number II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 246 URL: https://leetcode.ca/2016-08-02-246-Strobogrammatic-Number/ Title: Strobogrammatic Number Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 245 URL: https://leetcode.ca/2016-08-01-245-Shortest-Word-Distance-III/ Title: Shortest Word Distance III Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 244 URL: https://leetcode.ca/2016-07-31-244-Shortest-Word-Distance-II/ Title: Shortest Word Distance II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 243 URL: https://leetcode.ca/2016-07-30-243-Shortest-Word-Distance/ Title: Shortest Word Distance Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 186 URL: https://leetcode.ca/2016-06-03-186-Reverse-Words-in-a-String-II/ Title: Reverse Words in a String II Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 163 URL: https://leetcode.ca/2016-05-11-163-Missing-Ranges/ Title: Missing Ranges Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 161 URL: https://leetcode.ca/2016-05-09-161-One-Edit-Distance/ Title: One Edit Distance Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==

Problem Number: 159 URL: https://leetcode.ca/2016-05-07-159-Longest-Substring-with-At-Most-Two-Distinct-Characters/ Title: Longest Substring with At Most Two Distinct Characters Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==
Problem Number: 156 URL: https://leetcode.ca/2016-05-04-156-Binary-Tree-Upside-Down/ Title: Binary Tree Upside Down Problem Description: You are given a 2D integer array coordinates and an integer k, where coordinates[i] = [xi, yi] are the coordinates of the ith point in a 2D plane. We define the distance between two points (x1, y1) and (x2, y2) as (x1 XOR x2) + (y1 XOR y2) where XOR is the bitwise XOR operation. Return the number of pairs (i, j) such that i < j and the distance between points i and j is equal to k. Example 1:

Input: coordinates = [[1,2],[4,2],[1,3],[5,2]], k = 5 Output: 2 Explanation: We can choose the following pairs: - (0,1): Because we have (1 XOR 4) + (2 XOR 2) = 5. - (2,3): Because we have (1 XOR 5) + (3 XOR 2) = 5.

Example 2:

Input: coordinates = [[1,3],[1,3],[1,3],[1,3],[1,3]], k = 0 Output: 10 Explanation: Any two chosen pairs will have a distance of 0. There are 10 ways to choose two pairs.

Constraints:

2 <= coordinates.length <= 50000 0 <= xi, yi <= 106 0 <= k <= 100

==