Leetcode Offline

Single Element in a Sorted Array

Given a sorted array consisting of only integers where every element appears twice except for one element which appears once. Find this single element that appears only once.

Example 1:

Input: [1,1,2,3,3,4,4,8,8]
Output: 2

Example 2:

Input: [3,3,7,7,10,11,11]
Output: 10

Note:
Your solution should run in O(log n) time and O(1) space.

Coin Change 2

You are given coins of different denominations and a total amount of money. Write a function to compute the number of combinations that make up that amount. You may assume that you have infinite number of each kind of coin.

Note:
You can assume that

 0 <= amount <= 5000
 1 <= coin <= 5000
 the number of coins is less than 500
 the answer is guaranteed to fit into 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

Largest Palindrome Product

Find the largest palindrome made from the product of two n-digit numbers.
 Since the result could be very large, you should return the largest palindrome mod 1337.

Example:
Input: 2
Output: 987
Explanation: 99 x 91 = 9009, 9009 % 1337 = 987

Note:
The range of n is [1,8].

Find All Duplicates in an Array

Given an array of integers, 1 ≤ a[i] ≤ n (n = size of array), some elements appear twice and others appear once.

Find all the elements that appear twice in this array.

Could you do it without extra space and in O(n) runtime?

Example:

Input:
[4,3,2,7,8,2,3,1]

Output:
[2,3]

Two Sum

Given an array of integers, return indices of the two numbers such that they add up to a specific target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

Example:

Given nums = [2, 7, 11, 15], target = 9,

Because nums[0] + nums[1] = 2 + 7 = 9,
return [0, 1].

Add Two Numbers

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 contain a single digit. Add the two numbers and return it as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 0 -> 8

Longest Substring Without Repeating Characters

Given a string, find the length of the longest substring without repeating characters.

Examples:

Given "abcabcbb", the answer is "abc", which the length is 3.

Given "bbbbb", the answer is "b", with the length of 1.

Given "pwwkew", the answer is "wke", with the length of 3. Note that

the answer must be a substring, "pwke" is a subsequence and not a substring.

Median of Two Sorted Arrays

There are two sorted arrays nums1 and nums2 of size m and n respectively.

Find the median of the two sorted arrays. The overall run time complexity should be O(log (m+n)).

Example 1:

```
nums1 = [1, 3]
nums2 = [2]
```

The median is 2.0

Example 2:

```
nums1 = [1, 2]
nums2 = [3, 4]
```

The median is (2 + 3)/2 = 2.5

Longest Palindromic Substring

Given a string s, find the longest palindromic substring in s. You may assume that the maximum length of s is 1000.

Example:

Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.

Example:

Input: "cbbd"

Output: "bb"

ZigZag Conversion

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 text, int nRows);
```

convert("PAYPALISHIRING", 3) should return "PAHNAPLSIIGYIR".

Reverse Integer

Reverse digits of an integer.

Example1: x = 123, return 321
Example2: x = -123, return -321

click to show spoilers.

Have you thought about this?

Here are some good questions to ask before coding. Bonus points for you if you have already thought through this!

If the integer's last digit is 0, what should the output be? ie, cases such as 10, 100.

Did you notice that the reversed integer might overflow? Assume the input is a 32-bit integer, then the reverse of 1000000003 overflows. How should you handle such cases?

For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

Note:
The input is assumed to be a 32-bit signed integer. Your function should return 0 when the reversed integer overflows.

String to Integer (atoi)

Implement atoi to convert a string to an integer.

Hint: Carefully consider all possible input cases. If you want a challenge, please do not see below and ask yourself what are the possible input cases.

Notes:
It is intended for this problem to be specified vaguely (ie, no given input specs). You are responsible to gather all the input requirements up front.

Update (2015-02-10):
The signature of the C++ function had been updated. If you still see your function signature accepts a const char * argument, please click the reload button to reset your code definition.

spoilers alert... click to show requirements for atoi.

Requirements for atoi:

The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in str is not a valid integral number, or if no such sequence exists because either str is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned. If the correct value is out of the range of representable values, INT_MAX (2147483647) or INT_MIN (-2147483648) is returned.

Palindrome Number

Determine whether an integer is a palindrome. Do this without extra space.

click to show spoilers.

Some hints:

Could negative integers be palindromes? (ie, -1)

If you are thinking of converting the integer to string, note the restriction of using extra space.

You could also try reversing an integer. However, if you have solved the problem "Reverse Integer", you know that the reversed integer might overflow. How would you handle such case?

There is a more generic way of solving this problem.

Regular Expression Matching

Implement regular expression matching with support for '.' and '*'.

'.' Matches any single character.
'*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

The function prototype should be:
bool isMatch(const char *s, const char *p)

Some examples:
isMatch("aa","a") → false
isMatch("aa","aa") → true
isMatch("aaa","aa") → false
isMatch("aa", "a*") → true
isMatch("aa", ".*") → true
isMatch("ab", ".*") → true
isMatch("aab", "c*a*b") → true

Container With Most Water

Given n non-negative integers a1, a2, ..., an, where each represents a point at coordinate (i, ai). n vertical lines are drawn such that the two endpoints of line i is at (i, ai) and (i, 0). Find two lines, which together with x-axis forms a container, such that the container contains the most water.

Note: You may not slant the container and n is at least 2.

Integer to Roman

Given an integer, convert it to a roman numeral.

Input is guaranteed to be within the range from 1 to 3999.

Roman to Integer

Given a roman numeral, convert it to an integer.

Input is guaranteed to be within the range from 1 to 3999.

Longest Common Prefix

Write a function to find the longest common prefix string amongst an array of strings.

3Sum

Given an array S of n integers, are there elements a, b, c in S such that a + b + c = 0? Find all unique triplets in the array which gives the sum of zero.

Note: The solution set must not contain duplicate triplets.

For example, given array S = [-1, 0, 1, 2, -1, -4],

A solution set is:
[
  [-1, 0, 1],
  [-1, -1, 2]
]

3Sum Closest

Given an array S of n integers, find three integers in S such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

For example, given array S = {-1 2 1 -4}, and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

Letter Combinations of a Phone Number

Given a digit string, return all possible letter combinations that the number could represent.

A mapping of digit to letters (just like on the telephone buttons) is given below.

Input:Digit string "23"
Output: ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

Note:
Although the above answer is in lexicographical order, your answer could be in any order you want.

4Sum

Given an array S of n integers, are there elements a, b, c, and d in S such that a + b + c + d = target? Find all unique quadruplets in the array which gives the sum of target.

Note: The solution set must not contain duplicate quadruplets.

For example, given array S = [1, 0, -1, 0, -2, 2], and target = 0.

A solution set is:
```
[
  [-1,  0, 0, 1],
  [-2, -1, 1, 2],
  [-2,  0, 0, 2]
]
```


Remove Nth Node From End of List

Given a linked list, remove the nth node from the end of list and return its head.

For example,

Given linked list: 1->2->3->4->5, and n = 2.

After removing the second node from the end, the linked list becomes 1->2->3->5.

Note:
Given n will always be valid.
Try to do this in one pass.

Valid Parentheses

Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

The brackets must close in the correct order, "()" and "()[]{}" are all valid but "(]" and "([)]" are not.

Merge Two Sorted Lists

Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists.

Generate Parentheses

Given n pairs of parentheses, write a function to generate all combinations of well-formed parentheses.

For example, given n = 3, a solution set is:

```
[
  "((()))",
  "(()())",
  "(())()",
  "()(())",
  "()()()"
]
```


Merge k Sorted Lists

Merge k sorted linked lists and return it as one sorted list. Analyze and describe its complexity.

Swap Nodes in Pairs

Given a linked list, swap every two adjacent nodes and return its head.

For example,
Given 1->2->3->4, you should return the list as 2->1->4->3.

Your algorithm should use only constant space. You may not modify the values in the list, only nodes itself can be changed.

Reverse Nodes in k-Group

Given a linked list, reverse the nodes of a linked list k at a time and return its 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 nodes, only nodes itself may be changed.

Only constant memory is allowed.

For example,
Given this linked list: 1->2->3->4->5

For k = 2, you should return: 2->1->4->3->5

For k = 3, you should return: 3->2->1->4->5

Remove Duplicates from Sorted Array

Given a sorted array, remove the duplicates in place such that each element appear only once and return the new length.

Do not allocate extra space for another array, you must do this in place with constant memory.

For example,
Given input array nums = [1,1,2],

Your function should return length = 2, with the first two elements of nums being 1 and 2 respectively. It doesn't matter what you leave beyond the new length.

Remove Element

Given an array and a value, remove all instances of that value in place and return the new length.

Do not allocate extra space for another array, you must do this in place with constant memory.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

Example:
Given input array nums = [3,2,2,3], val = 3

Your function should return length = 2, with the first two elements of nums being 2.

  Try two pointers.
  Did you use the property of "the order of elements can be changed"?
  What happens when the elements to remove are rare?

Implement strStr()

Implement strStr().

Returns the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.

Divide Two Integers

Divide two integers without using multiplication, division and mod operator.

If it is overflow, return MAX_INT.

Substring with Concatenation of All Words

You are given a string, s, and a list of words, words, that are all of the same length. Find all starting indices of substring(s) in s that is a concatenation of each word in words exactly once and without any intervening characters.

For example, given:
s: "barfoothefoobarman"
words: ["foo", "bar"]

You should return the indices: [0,9].
(order does not matter).

Next Permutation

Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such arrangement is not possible, it must rearrange it as the lowest possible order (ie, sorted in ascending order).

The replacement must be in-place, do not allocate extra memory.

Here are some examples. Inputs are in the left-hand column and its corresponding outputs are in the right-hand column.
1,2,3 → 1,3,2
3,2,1 → 1,2,3
1,1,5 → 1,5,1

Longest Valid Parentheses

Given a string containing just the characters '(' and ')', find the length of the longest valid (well-formed) parentheses substring.

For "(()", the longest valid parentheses substring is "()", which has length = 2.

Another example is ")()())", where the longest valid parentheses substring is "()()", which has length = 4.

Search in Rotated Sorted Array

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

You are given a target value to search. If found in the array return its index, otherwise return -1.

You may assume no duplicate exists in the array.

Search for a Range

Given an array of integers sorted in ascending order, find the starting and ending position of a given target value.

Your algorithm's runtime complexity must be in the order of O(log n).

If the target is not found in the array, return [-1, -1].

For example,
Given [5, 7, 7, 8, 8, 10] and target value 8,
return [3, 4].

Search Insert Position

Given a sorted array 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 may assume no duplicates in the array.

Here are few examples.
[1,3,5,6], 5 → 2
[1,3,5,6], 2 → 1
[1,3,5,6], 7 → 4
[1,3,5,6], 0 → 0

Valid Sudoku

Determine if a Sudoku is valid, according to: Sudoku Puzzles - The Rules.

The Sudoku board could be partially filled, where empty cells are filled with the character '.'.

A partially filled sudoku which is valid.

Note:
A valid Sudoku board (partially filled) is not necessarily solvable. Only the filled cells need to be validated.

Sudoku Solver

Write a program to solve a Sudoku puzzle by filling the empty cells.

Empty cells are indicated by the character '.'.

You may assume that there will be only one unique solution.

A sudoku puzzle...

...and its solution numbers marked in red.

Count and Say

The count-and-say sequence is the sequence of integers beginning as follows:
1, 11, 21, 1211, 111221, ...

1 is read off as "one 1" or 11.
11 is read off as "two 1s" or 21.
21 is read off as "one 2, then one 1" or 1211.

Given an integer n, generate the nth sequence.

Note: The sequence of integers will be represented as a string.

Combination Sum

Given a set of candidate numbers (C) (without duplicates) and a target number (T), find all unique combinations in C where the candidate numbers sums to T.

The same repeated number may be chosen from C unlimited number of times.

Note:

All numbers (including target) will be positive integers.
The solution set must not contain duplicate combinations.

For example, given candidate set [2, 3, 6, 7] and target 7,
A solution set is:

```
[
  [7],
  [2, 2, 3]
]
```


Combination Sum II

Given a collection of candidate numbers (C) and a target number (T), find all unique combinations in C where the candidate numbers sums to T.

Each number in C may only be used once in the combination.

Note:

All numbers (including target) will be positive integers.
The solution set must not contain duplicate combinations.

For example, given candidate set [10, 1, 2, 7, 6, 1, 5] and target 8,
A solution set is:

```
[
  [1, 7],
  [1, 2, 5],
  [2, 6],
  [1, 1, 6]
]
```


First Missing Positive

Given an unsorted integer array, find the first missing positive integer.

For example,
Given [1,2,0] return 3,
and [3,4,-1,1] return 2.

Your algorithm should run in O(n) time and uses constant space.

Trapping Rain Water

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

For example,
Given [0,1,0,2,1,0,1,3,2,1,2,1], return 6.

The above elevation map 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. Thanks Marcos for contributing this image!

Multiply Strings

Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2.

Note:

The length of both num1 and num2 is < 110.
Both num1 and num2 contains only digits 0-9.
Both num1 and num2 does not contain any leading zero.
You must not use any built-in BigInteger library or convert the inputs to integer directly.

Wildcard Matching

Implement wildcard pattern matching with support for '?' and '*'.

'?' Matches any single character.
'*' Matches any sequence of characters (including the empty sequence).

The matching should cover the entire input string (not partial).

The function prototype should be:
bool isMatch(const char *s, const char *p)

Some examples:
isMatch("aa","a") → false
isMatch("aa","aa") → true

```
isMatch("aaa","aa") → false
isMatch("aa", "*") → true
isMatch("aa", "a*") → true
isMatch("ab", "?*") → true
isMatch("aab", "c*a*b") → false
```


Jump Game II

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Your goal is to reach the last index in the minimum number of jumps.

For example:
Given array A = [2,3,1,1,4]

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.)

Note:
You can assume that you can always reach the last index.

Permutations

Given a collection of distinct numbers, return all possible permutations.

For example,
[1,2,3] have the following permutations:

```
[
  [1,2,3],
  [1,3,2],
  [2,1,3],
  [2,3,1],
  [3,1,2],
  [3,2,1]
```

]

Permutations II

Given a collection of numbers that might contain duplicates, return all possible unique permutations.

For example,
[1,1,2] have the following unique permutations:

```
[
  [1,1,2],
  [1,2,1],
  [2,1,1]
]
```


Rotate Image

You are given an n x n 2D matrix representing an image.
Rotate the image by 90 degrees (clockwise).
Follow up:
Could you do this in-place?

Group Anagrams

Given an array of strings, group anagrams together.

For example, given: ["eat", "tea", "tan", "ate", "nat", "bat"],
Return:

```
[
  ["ate", "eat","tea"],
  ["nat","tan"],
  ["bat"]
]
```

Note: All inputs will be in lower-case.

Pow(x, n)

Implement pow(x, n).

N-Queens

The n-queens puzzle is the problem of placing n queens on an n×n chessboard such that no two queens attack each other.

Given an integer n, return all distinct solutions to the n-queens puzzle.

Each solution contains a distinct board configuration of the n-queens' placement, where 'Q' and '.' both indicate a queen and an empty space respectively.

For example,
There exist two distinct solutions to the 4-queens puzzle:

```
[
 [".Q..",  // Solution 1
  "...Q",
  "Q...",
  "..Q."],

 ["..Q.",  // Solution 2
  "Q...",
  "...Q",
  ".Q.."]
]
```


N-Queens II

Follow up for N-Queens problem.

Now, instead outputting board configurations, return the total number of distinct solutions.

Maximum Subarray

Find the contiguous subarray within an array (containing at least one number) which has the largest sum.

For example, given the array [-2,1,-3,4,-1,2,1,-5,4],
the contiguous subarray [4,-1,2,1] has the largest sum = 6.

click to show more practice.

More practice:

If you have figured out the O(n) solution, try coding another solution using the divide and conquer approach, which is more subtle.

Spiral Matrix

Given a matrix of m x n elements (m rows, n columns), return all elements of the matrix in spiral order.

For example,
Given the following matrix:

```
[
 [ 1, 2, 3 ],
 [ 4, 5, 6 ],
 [ 7, 8, 9 ]
]
```

You should return [1,2,3,6,9,8,7,4,5].

Jump Game

Given an array of non-negative integers, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Determine if you are able to reach the last index.

For example:
A = [2,3,1,1,4], return true.

A = [3,2,1,0,4], return false.

Merge Intervals

Given a collection of intervals, merge all overlapping intervals.

For example,
Given [1,3],[2,6],[8,10],[15,18],
return [1,6],[8,10],[15,18].

Insert Interval

Given a set of non-overlapping intervals, insert a new interval into the intervals (merge if necessary).

You may assume that the intervals were initially sorted according to their start times.

Example 1:
Given intervals [1,3],[6,9], insert and merge [2,5] in as [1,5], [6,9].

Example 2:
Given [1,2],[3,5],[6,7],[8,10],[12,16], insert and merge [4,9] in as [1,2],[3,10],[12,16].

This is because the new interval [4,9] overlaps with [3,5],[6,7], [8,10].

Length of Last Word

Given a string s consists of upper/lower-case alphabets and empty space characters ' ', return the length of last word in the string.

If the last word does not exist, return 0.

Note: A word is defined as a character sequence consists of non-space characters only.

For example,
Given s = "Hello World",
return 5.

Spiral Matrix II

Given an integer n, generate a square matrix filled with elements from 1 to n2 in spiral order.

For example,
Given n = 3,

You should return the following matrix:

```
[
 [ 1, 2, 3 ],
 [ 8, 9, 4 ],
 [ 7, 6, 5 ]
]
```


Permutation Sequence

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 (ie, for n = 3):

```
"123"
"132"
"213"
"231"
"312"
"321"
```

Given n and k, return the kth permutation sequence.

Note: Given n will be between 1 and 9 inclusive.

Rotate List

Given a list, rotate the list to the right by k places, where k is non-negative.

For example:
Given 1->2->3->4->5->NULL and k = 2,
return 4->5->1->2->3->NULL.

Unique Paths

A robot is located at the top-left corner of a m x n grid (marked 'Start' in the diagram below).

The robot can only move either down or right at any point in time. The robot is trying to reach the bottom-right corner of the grid (marked 'Finish' in the diagram below).

How many possible unique paths are there?

Above is a 3 x 7 grid. How many possible unique paths are there?

Note: m and n will be at most 100.

Unique Paths II

Follow up for "Unique Paths":

Now consider if some obstacles are added to the grids. How many unique paths would there be?

An obstacle and empty space is marked as 1 and 0 respectively in the grid.

For example,
There is one obstacle in the middle of a 3x3 grid as illustrated below.

```
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
```

The total number of unique paths is 2.

Note: m and n will be at most 100.

Minimum Path Sum

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.

Valid Number

Validate if a given string is numeric.

Some examples:
"0" => true
"   0.1 " => true
"abc" => false
"1 a" => false
"2e10" => true

Note: It is intended for the problem statement to be ambiguous. You should gather all requirements up front before implementing one.

Update (2015-02-10):
The signature of the C++ function had been updated. If you still see your function signature accepts a const char * argument, please click the reload button to reset your code definition.

Plus One

Given a non-negative integer represented as a non-empty array of digits, plus one to the integer.

You may assume the integer do not contain any leading zero, except the number 0 itself.

The digits are stored such that the most significant digit is at the head of the list.

Add Binary

Given two binary strings, return their sum (also a binary string).

For example,
a = "11"
b = "1"
Return "100".

Text Justification

Given an array of words and a length L, format the text such that each line has exactly L 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 L characters.

Extra spaces between words should be distributed as evenly as possible. If the number of spaces on a line do 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.

For example,
words: ["This", "is", "an", "example", "of", "text", "justification."]
L: 16.

Return the formatted lines as:

```
[
   "This    is    an",
   "example  of text",
   "justification.  "
]
```

Note: Each word is guaranteed not to exceed L in length.

click to show corner cases.

Corner Cases:

A line other than the last line might contain only one word. What should you do in this case?
In this case, that line should be left-justified.

Sqrt(x)

Implement int sqrt(int x).

Compute and return the square root of x.

Climbing Stairs

You are climbing a stair case. It takes n steps to reach to the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Note: Given n will be a positive integer.

Simplify Path

Given an absolute path for a file (Unix-style), simplify it.

For example,
path = "/home/", => "/home"
path = "/a/./b/../../c/", => "/c"

click to show corner cases.

Corner Cases:

Did you consider the case where path = "/../"?
In this case, you should return "/".
Another corner case is the path might contain multiple slashes '/' together, such as "/home//foo/".
In this case, you should ignore redundant slashes and return "/home/foo".

Edit Distance

Given two words word1 and word2, find the minimum number of steps required to convert word1 to word2. (each operation is counted as 1 step.)

You have the following 3 operations permitted on a word:

a) Insert a character
b) Delete a character
c) Replace a character

Set Matrix Zeroes

Given a m x n matrix, if an element is 0, set its entire row and column to 0. Do it in place.

click to show follow up.

Follow up:

Did you use extra space?
A straight forward 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?

Search a 2D Matrix

Write an efficient algorithm that searches for a value in an m x n matrix. This matrix has the following properties:

Integers in each row are sorted from left to right.
The first integer of each row is greater than the last integer of the previous row.

For example,

Consider the following matrix:

```
[
  [1,   3,  5,  7],
  [10, 11, 16, 20],
```

```
  [23, 30, 34, 50]
]
```

Given target = 3, return true.

Sort Colors

Given an array with n objects colored red, white or blue, sort them so that objects of the same color are adjacent, with the colors in the order red, white and blue.

Here, we will use the integers 0, 1, and 2 to represent the color red, white, and blue respectively.

Note:
You are not suppose to use the library's sort function for this problem.

click to show follow up.

Follow up:
A rather straight forward solution is a two-pass algorithm using counting sort.
First, iterate the array counting number of 0's, 1's, and 2's, then overwrite array with total number of 0's, then 1's and followed by 2's.
Could you come up with an one-pass algorithm using only constant space?

Minimum Window Substring

Given a string S and a string T, find the minimum window in S which will contain all the characters in T in complexity O(n).

For example,
S = "ADOBECODEBANC"
T = "ABC"

Minimum window is "BANC".

Note:
If there is no such window in S that covers all characters in T, return the empty string "".

If there are multiple such windows, you are guaranteed that there will always be only one unique minimum window in S.

Combinations

Given two integers n and k, return all possible combinations of k numbers out of 1 ... n.

For example,
If n = 4 and k = 2, a solution is:

```
[
  [2,4],
  [3,4],
  [2,3],
  [1,2],
  [1,3],
  [1,4],
]
```


Subsets

Given a set of distinct integers, nums, return all possible subsets.

Note: The solution set must not contain duplicate subsets.

For example,
If nums = [1,2,3], a solution is:

```
[
  [3],
  [1],
  [2],
```

```
  [1,2,3],
  [1,3],
  [2,3],
  [1,2],
  []
]
```


Word Search

Given a 2D board and a word, find if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cell, where "adjacent" cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example,
Given board =

```
[
  ['A','B','C','E'],
  ['S','F','C','S'],
  ['A','D','E','E']
]
```

word = "ABCCED", -> returns true,
word = "SEE", -> returns true,
word = "ABCB", -> returns false.

Remove Duplicates from Sorted Array II

Follow up for "Remove Duplicates":
What if duplicates are allowed at most twice?

For example,
Given sorted array nums = [1,1,1,2,2,3],

Your function should return length = 5, with the first five elements of nums being 1, 1, 2, 2 and 3. It doesn't matter what you leave beyond the new length.

Search in Rotated Sorted Array II

Follow up for "Search in Rotated Sorted Array":
What if duplicates are allowed?

Would this affect the run-time complexity? How and why?

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

Write a function to determine if a given target is in the array.

The array may contain duplicates.

Remove Duplicates from Sorted List II

Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list.

For example,
Given 1->2->3->3->4->4->5, return 1->2->5.
Given 1->1->1->2->3, return 2->3.

Remove Duplicates from Sorted List

Given a sorted linked list, delete all duplicates such that each element appear only once.

For example,
Given 1->1->2, return 1->2.
Given 1->1->2->3->3, return 1->2->3.

Largest Rectangle in Histogram

Given n non-negative integers representing the histogram's bar height where the width of each bar is 1, find the area of largest rectangle in the histogram.

Above is a histogram where width of each bar is 1, given height = [2,1,5,6,2,3].

The largest rectangle is shown in the shaded area, which has area = 10 unit.

For example,
Given heights = [2,1,5,6,2,3],
return 10.

Maximal Rectangle

Given a 2D binary matrix filled with 0's and 1's, find the largest rectangle containing only 1's and return its area.

For example, given the following matrix:

1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0

Return 6.

Partition List

Given 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.

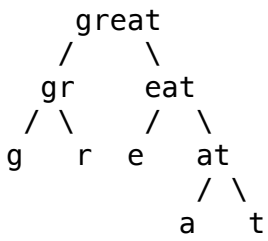
You should preserve the original relative order of the nodes in each of the two partitions.

For example,
Given 1->4->3->2->5->2 and x = 3,
return 1->2->2->4->3->5.

Scramble String

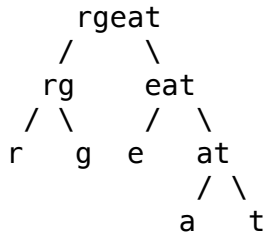
Given a string s1, we may represent it as a binary tree by partitioning it to two non-empty substrings recursively.

Below is one possible representation of s1 = "great":



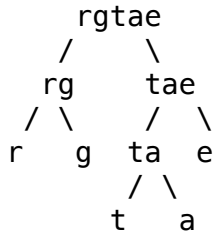
To scramble the string, we may choose any non-leaf node and swap its two children.

For example, if we choose the node "gr" and swap its two children, it produces a scrambled string "rgeat".



We say that "rgeat" is a scrambled string of "great".

Similarly, if we continue to swap the children of nodes "eat" and "at", it produces a scrambled string "rgtae".



We say that "rgtae" is a scrambled string of "great".

Given two strings s1 and s2 of the same length, determine if s2 is a

scrambled string of s1.

Merge Sorted Array

Given two sorted integer arrays nums1 and nums2, merge nums2 into nums1 as one sorted array.

Note:
You may assume that nums1 has enough space (size that is greater or equal to m + n) to hold additional elements from nums2. The number of elements initialized in nums1 and nums2 are m and n respectively.

Gray Code

The gray code is a binary numeral system where two successive values differ in only one bit.

Given a non-negative integer n representing the total number of bits in the code, print the sequence of gray code. A gray code sequence must begin with 0.

For example, given n = 2, return [0,1,3,2]. Its gray code sequence is:

00 - 0
01 - 1
11 - 3
10 - 2

Note:
For a given n, a gray code sequence is not uniquely defined.

For example, [0,2,3,1] is also a valid gray code sequence according to the above definition.

For now, the judge is able to judge based on one instance of gray code sequence. Sorry about that.

Subsets II

Given a collection of integers that might contain duplicates, nums, return all possible subsets.

Note: The solution set must not contain duplicate subsets.

For example,

If nums = [1,2,2], a solution is:

```
[
  [2],
  [1],
  [1,2,2],
  [2,2],
  [1,2],
  []
]
```

Decode Ways

A message containing letters from A-Z is being encoded to numbers using the following mapping:

```
'A' -> 1
'B' -> 2
...
'Z' -> 26
```

Given an encoded message containing digits, determine the total number of ways to decode it.

For example,
Given encoded message "12",
it could be decoded as "AB" (1 2) or "L" (12).

The number of ways decoding "12" is 2.

Reverse Linked List II

Reverse a linked list from position m to n. Do it in-place and in one-pass.

For example:

Given 1->2->3->4->5->NULL, m = 2 and n = 4,

return 1->4->3->2->5->NULL.

Note:
Given m, n satisfy the following condition:
1 ≤ m ≤ n ≤ length of list.

Restore IP Addresses

Given a string containing only digits, restore it by returning all possible valid IP address combinations.

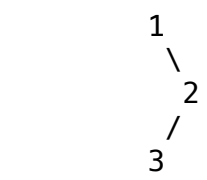
For example:
Given "25525511135",

return ["255.255.11.135", "255.255.111.35"]. (Order does not matter)

Binary Tree Inorder Traversal

Given a binary tree, return the inorder traversal of its nodes' values.

For example:
Given binary tree [1,null,2,3],



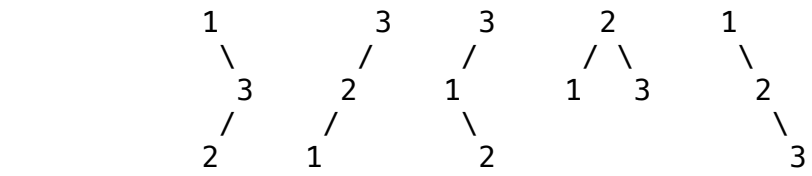
return [1,3,2].

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

Unique Binary Search Trees II

Given an integer n, generate all structurally unique BST's (binary search trees) that store values 1...n.

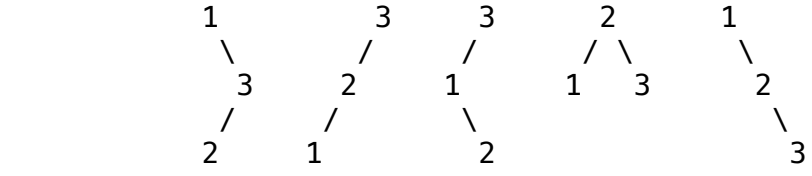
For example,
Given n = 3, your program should return all 5 unique BST's shown below.



Unique Binary Search Trees

Given n, how many structurally unique BST's (binary search trees) that store values 1...n?

For example,
Given n = 3, there are a total of 5 unique BST's.



Interleaving String

Given s1, s2, s3, find whether s3 is formed by the interleaving of s1 and s2.

For example,
Given:
s1 = "aabcc",
s2 = "dbbca",

When s3 = "aadbbcbcac", return true.
When s3 = "aadbbbaccc", return false.

Validate Binary Search Tree

Given a binary tree, determine if it is a valid 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:

```
    2
   / \
  1   3
```

Binary tree [2,1,3], return true.

Example 2:

```
    1
   / \
  2   3
```

Binary tree [1,2,3], return false.

Recover Binary Search Tree

Two elements of a binary search tree (BST) are swapped by mistake.

Recover the tree without changing its structure.

Note:
A solution using O(n) space is pretty straight forward. Could you devise a constant space solution?

Same Tree

Given two binary trees, write a function to check if they are equal

or not.

Two binary trees are considered equal if they are structurally identical and the nodes have the same value.

Symmetric Tree

Given a binary tree, check whether it is a mirror of itself (ie, symmetric around its center).

For example, this binary tree [1,2,2,3,4,4,3] is symmetric:

```
    1
   / \
  2   2
 / \ / \
3  4 4  3
```

But the following [1,2,2,null,3,null,3] is not:

```
    1
   / \
  2   2
   \   \
   3    3
```

Note:
Bonus points if you could solve it both recursively and iteratively.

Binary Tree Level Order Traversal

Given a binary tree, return the level order traversal of its nodes' values. (ie, from left to right, level by level).

For example:
Given binary tree [3,9,20,null,null,15,7],

```
    3
   / \
  9  20
    /  \
   15   7
```

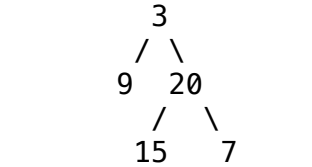
return its level order traversal as:

```
[
  [3],
  [9,20],
  [15,7]
]
```


Binary Tree Zigzag Level Order Traversal

Given a binary tree, return the zigzag level order traversal of its nodes' values. (ie, from left to right, then right to left for the next level and alternate between).

For example:
Given binary tree [3,9,20,null,null,15,7],



return its zigzag level order traversal as:

```
[
  [3],
  [20,9],
  [15,7]
]
```


Maximum Depth of Binary Tree

Given a binary 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.

Construct Binary Tree from Preorder and Inorder Traversal

Given preorder and inorder traversal of a tree, construct the binary

tree.

Note:
You may assume that duplicates do not exist in the tree.

Construct Binary Tree from Inorder and Postorder Traversal

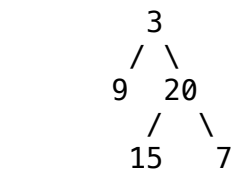
Given inorder and postorder traversal of a tree, construct the binary tree.

Note:
You may assume that duplicates do not exist in the tree.

Binary Tree Level Order Traversal II

Given a binary tree, return the bottom-up level order traversal of its nodes' values. (ie, from left to right, level by level from leaf to root).

For example:
Given binary tree [3,9,20,null,null,15,7],



return its bottom-up level order traversal as:

```
[
  [15,7],
  [9,20],
  [3]
]
```


Convert Sorted Array to Binary Search Tree

Given an array where elements are sorted in ascending order, convert it to a height balanced BST.

Convert Sorted List to Binary Search Tree

Given a singly linked list where elements are sorted in ascending order, convert it to a height balanced BST.

Balanced Binary Tree

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

Minimum Depth of Binary Tree

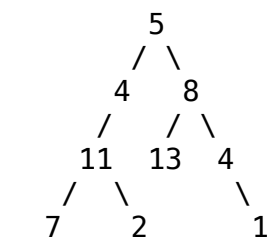
Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

Path Sum

Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the given sum.

For example:
Given the below binary tree and sum = 22,

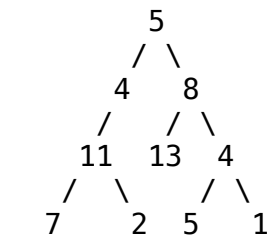


return true, as there exist a root-to-leaf path 5->4->11->2 which sum is 22.

Path Sum II

Given a binary tree and a sum, find all root-to-leaf paths where each path's sum equals the given sum.

For example:
Given the below binary tree and sum = 22,



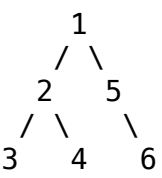
return

```
[
   [5,4,11,2],
   [5,8,4,5]
]
```

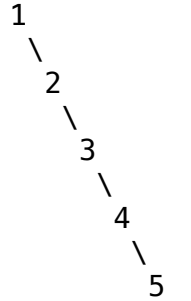

Flatten Binary Tree to Linked List

Given a binary tree, flatten it to a linked list in-place.

For example,
Given



The flattened tree should look like:



```
            \
             6
```

click to show hints.

Hints:
If you notice carefully in the flattened tree, each node's right child points to the next node of a pre-order traversal.

Distinct Subsequences

Given a string S and a string T, count the number of distinct subsequences of T in S.

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, "ACE" is a subsequence of "ABCDE" while "AEC" is not).

Here is an example:
S = "rabbbit", T = "rabbit"

Return 3.

Populating Next Right Pointers in Each Node

Given a binary tree

```
    struct TreeLinkNode {
      TreeLinkNode *left;
      TreeLinkNode *right;
      TreeLinkNode *next;
    }
```

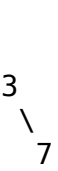
Populate each next pointer to point to its next right node. If there is no next right node, the next pointer should be set to NULL.

Initially, all next pointers are set to NULL.

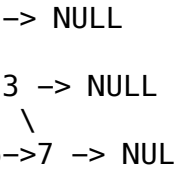
Note:

You may only use constant extra space.
You may assume that it is a perfect binary tree (ie, all leaves are at the same level, and every parent has two children).

For example,
Given the following perfect binary tree,



After calling your function, the tree should look like:

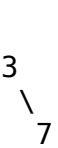


Populating Next Right Pointers in Each Node II

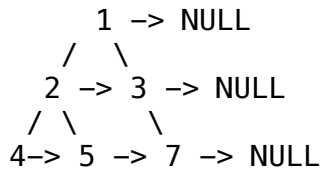
Follow up for problem "Populating Next Right Pointers in Each Node". What if the given tree could be any binary tree? Would your previous solution still work?

Note:
You may only use constant extra space.

For example,
Given the following binary tree,



After calling your function, the tree should look like:



Pascal's Triangle

Given numRows, generate the first numRows of Pascal's triangle.

For example, given numRows = 5,
Return

```
[
     [1],
    [1,1],
   [1,2,1],
  [1,3,3,1],
 [1,4,6,4,1]
]
```


Pascal's Triangle II

Given an index k, return the kth row of the Pascal's triangle.

For example, given k = 3,
Return [1,3,3,1].

Note:
Could you optimize your algorithm to use only O(k) extra space?

Triangle

Given a triangle, find the minimum path sum from top to bottom. Each step you may move to adjacent numbers on the row below.

For example, given the following triangle

```
[
     [2],
```

```
    [3,4],
   [6,5,7],
  [4,1,8,3]
]
```

The minimum path sum from top to bottom is 11 (i.e., 2 + 3 + 5 + 1 = 11).

Note:
Bonus point if you are able to do this using only O(n) extra space, where n is the total number of rows in the triangle.

Best Time to Buy and Sell Stock

Say you have an array for which the ith element is the price of a given stock on day i.

If you were only permitted to complete at most one transaction (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.

Example 1:

Input: [7, 1, 5, 3, 6, 4]
Output: 5

max. difference = 6-1 = 5 (not 7-1 = 6, as selling price needs to be larger than buying price)

Example 2:

Input: [7, 6, 4, 3, 1]
Output: 0

In this case, no transaction is done, i.e. max profit = 0.

Best Time to Buy and Sell Stock II

Say you have an array for which the ith element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the

stock multiple times). However, you may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

Best Time to Buy and Sell Stock III

Say you have an array for which the ith element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete at most two transactions.

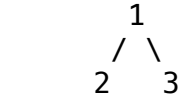
Note:
You may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

Binary Tree Maximum Path Sum

Given a binary tree, find the maximum path sum.

For this problem, a path is defined as any sequence of nodes from some starting node to any node in the tree along the parent-child connections. The path must contain at least one node and does not need to go through the root.

For example:
Given the below binary tree,



Return 6.

Valid Palindrome

Given a string, determine if it is a palindrome, considering only alphanumeric characters and ignoring cases.

For example,
"A man, a plan, a canal: Panama" is a palindrome.
"race a car" is not a palindrome.

Note:
Have you consider that the string might be empty? This is a good question to ask during an interview.

For the purpose of this problem, we define empty string as valid palindrome.

Word Ladder II

Given two words (beginWord and endWord), and a dictionary's word list, find all shortest transformation sequence(s) from beginWord to endWord, such that:

Only one letter can be changed at a time
Each transformed word must exist in the word list. Note that beginWord is not a transformed word.

For example,

Given:
beginWord = "hit"
endWord = "cog"
wordList = ["hot","dot","dog","lot","log","cog"]

Return

```
  [
    ["hit","hot","dot","dog","cog"],
    ["hit","hot","lot","log","cog"]
  ]
```


Note:

Return an empty list if there is no such transformation sequence.
All words have the same length.
All words contain only lowercase alphabetic characters.
You may assume no duplicates in the word list.
You may assume beginWord and endWord are non-empty and are not the same.

UPDATE (2017/1/20):
The wordList parameter had been changed to a list of strings (instead of a set of strings). Please reload the code definition to get the latest changes.

Word Ladder

Given two words (beginWord and endWord), and a dictionary's word list, find the length of shortest transformation sequence from beginWord to endWord, such that:

Only one letter can be changed at a time.
Each transformed word must exist in the word list. Note that beginWord is not a transformed word.

For example,

Given:
beginWord = "hit"
endWord = "cog"
wordList = ["hot","dot","dog","lot","log","cog"]

As one shortest transformation is "hit" -> "hot" -> "dot" -> "dog" -> "cog",
return its length 5.

Note:

Return 0 if there is no such transformation sequence.
All words have the same length.
All words contain only lowercase alphabetic characters.
You may assume no duplicates in the word list.
You may assume beginWord and endWord are non-empty and are not the same.

UPDATE (2017/1/20):
The wordList parameter had been changed to a list of strings (instead of a set of strings). Please reload the code definition to

get the latest changes.

Longest Consecutive Sequence

Given an unsorted array of integers, find the length of the longest consecutive elements sequence.

For example,
Given [100, 4, 200, 1, 3, 2],
The longest consecutive elements sequence is [1, 2, 3, 4]. Return its length: 4.

Your algorithm should run in O(n) complexity.

Sum Root to Leaf Numbers

Given a binary tree containing digits from 0-9 only, each root-to-leaf path could represent a number.
An example is the root-to-leaf path 1->2->3 which represents the number 123.

Find the total sum of all root-to-leaf numbers.

For example,

```
    1
   / \
  2   3
```

The root-to-leaf path 1->2 represents the number 12.
The root-to-leaf path 1->3 represents the number 13.

Return the sum = 12 + 13 = 25.

Surrounded Regions

Given a 2D board containing 'X' and 'O' (the letter O), capture all regions surrounded by 'X'.

A region is captured by flipping all 'O's into 'X's in that surrounded region.

For example,

X X X X
X O O X
X X O X
X O X X

After running your function, the board should be:

X X X X
X X X X
X X X X
X O X X

Palindrome Partitioning

Given a string s, partition s such that every substring of the partition is a palindrome.

Return all possible palindrome partitioning of s.

For example, given s = "aab",

Return

```
[
  ["aa","b"],
  ["a","a","b"]
]
```


Palindrome Partitioning II

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.

For example, given s = "aab",
Return 1 since the palindrome partitioning ["aa","b"] could be produced using 1 cut.

Clone Graph

Clone an undirected graph. Each node in the graph contains a label and a list of its neighbors.

OJ's undirected graph serialization:

Nodes are labeled uniquely.

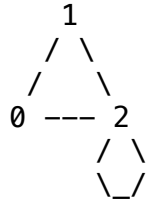
We use # as a separator for each node, and , as a separator for node label and each neighbor of the node.

As an example, consider the serialized graph {0,1,2#1,2#2,2}.

The graph has a total of three nodes, and therefore contains three parts as separated by #.

First node is labeled as 0. Connect node 0 to both nodes 1 and 2.
Second node is labeled as 1. Connect node 1 to node 2.
Third node is labeled as 2. Connect node 2 to node 2 (itself), thus forming a self-cycle.

Visually, the graph looks like the following:



Gas Station

There are N gas stations along a circular route, where the amount of gas at station i is gas[i].

You have a car with an unlimited gas tank and it costs cost[i] of gas to travel from station i to its next station (i+1). You begin the journey with an empty tank at one of the gas stations.

Return the starting gas station's index if you can travel around the circuit once, otherwise return -1.

Note:
The solution is guaranteed to be unique.

Candy

There are N children standing in a line. Each child is assigned a rating value.

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.

What is the minimum candies you must give?

Single Number

Given an array of integers, every element appears twice except for one. Find that single one.

Note:
Your algorithm should have a linear runtime complexity. Could you

implement it without using extra memory?

Single Number II

Given an array of integers, every element appears three times except for one, which appears exactly once. Find that single one.

Note:
Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

Copy List with Random Pointer

A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null.

Return a deep copy of the list.

Word Break

Given a non-empty string s and a dictionary wordDict containing a list of non-empty words, determine if s can be segmented into a space-separated sequence of one or more dictionary words. You may assume the dictionary does not contain duplicate words.

For example, given
s = "leetcode",
dict = ["leet", "code"].

Return true because "leetcode" can be segmented as "leet code".

UPDATE (2017/1/4):
The wordDict parameter had been changed to a list of strings (instead of a set of strings). Please reload the code definition to get the latest changes.

Word Break II

Given a non-empty string s and a dictionary wordDict containing a list of non-empty words, add spaces in s to construct a sentence where each word is a valid dictionary word. You may assume the dictionary does not contain duplicate words.

Return all such possible sentences.

For example, given
s = "catsanddog",
dict = ["cat", "cats", "and", "sand", "dog"].

A solution is ["cats and dog", "cat sand dog"].

UPDATE (2017/1/4):
The wordDict parameter had been changed to a list of strings (instead of a set of strings). Please reload the code definition to get the latest changes.

Linked List Cycle

Given a linked list, determine if it has a cycle in it.

Follow up:
Can you solve it without using extra space?

Linked List Cycle II

Given a linked list, return the node where the cycle begins. If there is no cycle, return null.

Note: Do not modify the linked list.

Follow up:
Can you solve it without using extra space?

Reorder List

Given a singly linked list L: L0→L1→…→Ln-1→Ln,
reorder it to: L0→Ln→L1→Ln-1→L2→Ln-2→…

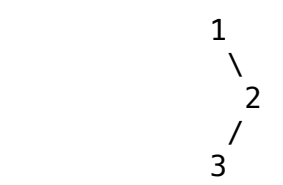
You must do this in-place without altering the nodes' values.

For example,
Given {1,2,3,4}, reorder it to {1,4,2,3}.

Binary Tree Preorder Traversal

Given a binary tree, return the preorder traversal of its nodes' values.

For example:
Given binary tree {1,#,2,3},



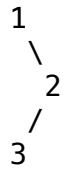
return [1,2,3].

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

Binary Tree Postorder Traversal

Given a binary tree, return the postorder traversal of its nodes' values.

For example:
Given binary tree {1,#,2,3},



return [3,2,1].

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

LRU Cache

Design and implement a data structure for Least Recently Used (LRU) cache. It should support the following operations: get and put.

get(key) - Get the value (will always be positive) of the key if the key exists in the cache, otherwise return -1.
put(key, value) - Set or insert the value if the key is not already present. When the cache reached its capacity, it should invalidate the least recently used item before inserting a new item.

Follow up:
Could you do both operations in O(1) time complexity?

Example:

```
LRUCache cache = new LRUCache( 2 /* capacity */ );

cache.put(1, 1);
cache.put(2, 2);
cache.get(1);       // returns 1
cache.put(3, 3);    // evicts key 2
cache.get(2);       // returns -1 (not found)
cache.put(4, 4);    // evicts key 1
cache.get(1);       // returns -1 (not found)
cache.get(3);       // returns 3
cache.get(4);       // returns 4
```

Insertion Sort List

Sort a linked list using insertion sort.

Sort List

Sort a linked list in O(n log n) time using constant space complexity.

Max Points on a Line

Given n points on a 2D plane, find the maximum number of points that lie on the same straight line.

Evaluate Reverse Polish Notation

Evaluate the value of an arithmetic expression in Reverse Polish Notation.

Valid operators are +, -, *, /. Each operand may be an integer or another expression.

Some examples:

```
  ["2", "1", "+", "3", "*"] -> ((2 + 1) * 3) -> 9
  ["4", "13", "5", "/", "+"] -> (4 + (13 / 5)) -> 6
```


Reverse Words in a String

Given an input string, reverse the string word by word.

For example,
Given s = "the sky is blue",
return "blue is sky the".

Update (2015-02-12):
For C programmers: Try to solve it in-place in O(1) space.

click to show clarification.

Clarification:

What constitutes a word?
A sequence of non-space characters constitutes a word.
Could the input string contain leading or trailing spaces?
Yes. However, your reversed string should not contain leading or trailing spaces.
How about multiple spaces between two words?
Reduce them to a single space in the reversed string.

Maximum Product Subarray

Find the contiguous subarray within an array (containing at least one number) which has the largest product.

For example, given the array [2,3,-2,4],
the contiguous subarray [2,3] has the largest product = 6.

Find Minimum in Rotated Sorted Array

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

Find the minimum element.

You may assume no duplicate exists in the array.

Find Minimum in Rotated Sorted Array II

Follow up for "Find Minimum in Rotated Sorted Array":
What if duplicates are allowed?

Would this affect the run-time complexity? How and why?

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., 0 1 2 4 5 6 7 might become 4 5 6 7 0 1 2).

Find the minimum element.

The array may contain duplicates.

Min Stack

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

push(x) -- Push element x onto stack.

pop() -- Removes the element on top of the stack.

top() -- Get the top element.

getMin() -- Retrieve the minimum element in the stack.

Example:

```
MinStack minStack = new MinStack();
minStack.push(-2);
minStack.push(0);
minStack.push(-3);
minStack.getMin();   --> Returns -3.
minStack.pop();
minStack.top();      --> Returns 0.
minStack.getMin();   --> Returns -2.
```


Intersection of Two Linked Lists

Write a program to find the node at which the intersection of two singly linked lists begins.

For example, the following two linked lists:

```
A:          a1 → a2
                   ↘
                     c1 → c2 → c3
                   ↗
```

B: b1 → b2 → b3

begin to intersect at node c1.

Notes:

If the two linked lists have no intersection at all, return null.
The linked lists must retain their original structure after the function returns.
You may assume there are no cycles anywhere in the entire linked structure.
Your code should preferably run in O(n) time and use only O(1) memory.

Credits:Special thanks to @stellari for adding this problem and creating all test cases.

Find Peak Element

A peak element is an element that is greater than its neighbors.

Given an input array where num[i] ≠ num[i+1], find a peak element and return its index.

The array may contain multiple peaks, in that case return the index to any one of the peaks is fine.

You may imagine that num[-1] = num[n] = -∞.

For example, in array [1, 2, 3, 1], 3 is a peak element and your function should return the index number 2.

click to show spoilers.

Note:
Your solution should be in logarithmic complexity.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Maximum Gap

Given an unsorted array, find the maximum difference between the successive elements in its sorted form.

Try to solve it in linear time/space.

Return 0 if the array contains less than 2 elements.

You may assume all elements in the array are non-negative integers and fit in the 32-bit signed integer range.

Credits:Special thanks to @porker2008 for adding this problem and creating all test cases.

Compare Version Numbers

Compare two version numbers version1 and version2.
If version1 > version2 return 1, if version1 < version2 return -1, otherwise return 0.

You may assume that the version strings are non-empty and contain only digits and the . character.
The . character does not represent a decimal point and is used to separate number sequences.
For instance, 2.5 is not "two and a half" or "half way to version three", it is the fifth second-level revision of the second first-level revision.

Here is an example of version numbers ordering:
0.1 < 1.1 < 1.2 < 13.37

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Fraction to Recurring Decimal

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.

For example,

Given numerator = 1, denominator = 2, return "0.5".
Given numerator = 2, denominator = 1, return "2".
Given numerator = 2, denominator = 3, return "0.(6)".

No scary math, just apply elementary math knowledge. Still remember how to perform a long division?

Try a long division on 4/9, the repeating part is obvious. Now try 4/333. Do you see a pattern?

Be wary of edge cases! List out as many test cases as you can think of and test your code thoroughly.

Credits:Special thanks to @Shangrila for adding this problem and creating all test cases.

Two Sum II - Input array is sorted

Given an array of integers that is already sorted in ascending order, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have exactly one solution and you may not use the same element twice.

Input: numbers={2, 7, 11, 15}, target=9
Output: index1=1, index2=2

Excel Sheet Column Title

Given a positive integer, return its corresponding column title as appear in an Excel sheet.

For example:

    1 -> A
    2 -> B
    3 -> C
    ...
    26 -> Z
    27 -> AA
    28 -> AB

Credits:Special thanks to @ifanchu for adding this problem and creating all test cases.

Majority Element

Given an array of size n, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

Credits:Special thanks to @ts for adding this problem and creating

all test cases.

Excel Sheet Column Number

Related to question Excel Sheet Column Title
Given a column title as appear in an Excel sheet, return its corresponding column number.

For example:
    A -> 1
    B -> 2
    C -> 3
    ...
    Z -> 26
    AA -> 27
    AB -> 28

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Factorial Trailing Zeroes

Given an integer n, return the number of trailing zeroes in n!.

Note: Your solution should be in logarithmic time complexity.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Binary Search Tree Iterator

Implement an iterator over a binary search tree (BST). Your iterator will be initialized with the root node of a BST.

Calling next() will return the next smallest number in the BST.

Note: next() and hasNext() should run in average O(1) time and uses O(h) memory, where h is the height of the tree.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Dungeon Game

```
table.dungeon, .dungeon th, .dungeon td {
  border:3px solid black;
}
```

```
 .dungeon th, .dungeon td {
    text-align: center;
    height: 70px;
    width: 70px;
}
```

The demons had captured the princess (P) 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 (K) was initially positioned in the top-left room and must fight his way through the 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, so the knight loses health (negative integers) upon entering these rooms;
other rooms are either empty (0's) or contain magic orbs that increase the knight's health (positive integers).
In order to reach the princess as quickly as possible, the knight decides to move only rightward or downward in each step.

Write a function to determine the knight's minimum initial health so that he is able to rescue the princess.
For example, given the dungeon below, the initial health of the knight must be at least 7 if he follows the optimal path RIGHT-> RIGHT -> DOWN -> DOWN.

-2 (K)
-3
3

-5
-10
1

10
30
-5 (P)

Notes:

The knight's health has no upper bound.
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.

Credits:Special thanks to @stellari for adding this problem and creating all test cases.

Largest Number

Given a list of non negative integers, arrange them such that they form the largest number.

For example, given [3, 30, 34, 5, 9], the largest formed number is 9534330.

Note: The result may be very large, so you need to return a string instead of an integer.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Repeated DNA Sequences

All DNA is composed of a series of nucleotides abbreviated as A, C, G, and T, for example: "ACGAATTCCG". When studying DNA, it is sometimes useful to identify repeated sequences within the DNA.

Write a function to find all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule.

For example,

Given s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT",

Return:
["AAAAACCCCC", "CCCCCAAAAA"].

Best Time to Buy and Sell Stock IV

Say you have an array for which the ith element is the price of a given stock on day i.

Design an algorithm to find the maximum profit. You may complete at most k transactions.

Note:
You may not engage in multiple transactions at the same time (ie,

you must sell the stock before you buy again).

Credits:Special thanks to @Freezen for adding this problem and creating all test cases.

Rotate Array

Rotate an array of n elements to the right by k steps.
For example, with n = 7 and k = 3, the array [1,2,3,4,5,6,7] is rotated to [5,6,7,1,2,3,4].

Note:
Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.

[show hint]
Hint:
Could you do it in-place with O(1) extra space?

Related problem: Reverse Words in a String II

Credits:Special thanks to @Freezen for adding this problem and creating all test cases.

Reverse Bits

Reverse bits of a given 32 bits unsigned integer.

For example, given input 43261596 (represented in binary as 00000010100101000001111010011100), return 964176192 (represented in binary as 00111001011110000010100101000000).

Follow up:
If this function is called many times, how would you optimize it?

Related problem: Reverse Integer

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Number of 1 Bits

Write a function that takes an unsigned integer and returns the number of ’1' bits it has (also known as the Hamming weight).

For example, the 32-bit integer ’11' has binary representation 00000000000000000000000000001011, so the function should return 3.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

House Robber

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 system connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

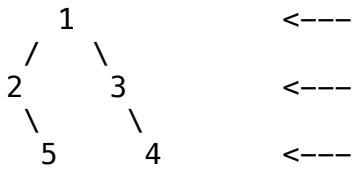
Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight without alerting the police.

Credits:Special thanks to @ifanchu for adding this problem and creating all test cases. Also thanks to @ts for adding additional test cases.

Binary Tree Right Side View

Given 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.

For example:
Given the following binary tree,



You should return [1, 3, 4].

Credits:Special thanks to @amrsaqr for adding this problem and creating all test cases.

Number of Islands

Given a 2d grid map of '1's (land) and '0's (water), count 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:
11110110101100000000
Answer: 1
Example 2:
11000110000010000011
Answer: 3

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

Bitwise AND of Numbers Range

Given a range [m, n] where 0 <= m <= n <= 2147483647, return the bitwise AND of all numbers in this range, inclusive.

For example, given the range [5, 7], you should return 4.

Credits:Special thanks to @amrsaqr for adding this problem and creating all test cases.

Happy Number

Write an algorithm to determine if a number 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, and 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 numbers.

Example: 19 is a happy number

12 + 92 = 82
82 + 22 = 68
62 + 82 = 100
12 + 02 + 02 = 1

Credits:Special thanks to @mithmatt and @ts for adding this problem and creating all test cases.

Remove Linked List Elements

Remove all elements from a linked list of integers that have value val.

Example
Given: 1 --> 2 --> 6 --> 3 --> 4 --> 5 --> 6, val = 6
Return: 1 --> 2 --> 3 --> 4 --> 5

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

Count Primes

Description:
Count the number of prime numbers less than a non-negative number, n.

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

  Let's start with a isPrime function. To determine if a number is prime, we need to check if it is not divisible by any number less than n. The runtime complexity of isPrime function would be O(n) and hence counting the total prime numbers up to n would be O(n2). Could we do better?

  As we know the number must not be divisible by any number > n / 2, we can immediately cut the total iterations half by dividing only up to n / 2. Could we still do better?

  Let's write down all of 12's factors:

2 × 6 = 12
3 × 4 = 12
4 × 3 = 12
6 × 2 = 12

As you can see, calculations of 4 × 3 and 6 × 2 are not necessary. Therefore, we only need to consider factors up to √n because, if n is divisible by some number p, then n = p × q and since p ≤ q, we could derive that p ≤ √n.

Our total runtime has now improved to O(n1.5), which is slightly better. Is there a faster approach?


```
public int countPrimes(int n) {
   int count = 0;
   for (int i = 1; i < n; i++) {
      if (isPrime(i)) count++;
   }
   return count;
}
```

```
private boolean isPrime(int num) {
   if (num <= 1) return false;
   // Loop's ending condition is i * i <= num instead of i <=
sqrt(num)
   // to avoid repeatedly calling an expensive function sqrt().
   for (int i = 2; i * i <= num; i++) {
      if (num % i == 0) return false;
   }
   return true;
}
```

The Sieve of Eratosthenes is one of the most efficient ways to find all prime numbers up to n. But don't let that name scare you, I promise that the concept is surprisingly simple.

Sieve of Eratosthenes: algorithm steps for primes below 121. "Sieve of Eratosthenes Animation" by SKopp is licensed under CC BY 2.0.

We start off with a table of n numbers. Let's look at the first number, 2. We know all multiples of 2 must not be primes, so we mark them off as non-primes. Then we look at the next number, 3. Similarly, all multiples of 3 such as 3 × 2 = 6, 3 × 3 = 9, ... must not be primes, so we mark them off as well. Now we look at the next number, 4, which was already marked off. What does this tell you? Should you mark off all multiples of 4 as well?

4 is not a prime because it is divisible by 2, which means all multiples of 4 must also be divisible by 2 and were already marked off. So we can skip 4 immediately and go to the next number, 5. Now, all multiples of 5 such as 5 × 2 = 10, 5 × 3 = 15, 5 × 4 = 20, 5 × 5 = 25, ... can be marked off. There is a slight optimization here, we do not need to start from 5 × 2 = 10. Where should we start marking off?

In fact, we can mark off multiples of 5 starting at 5 × 5 = 25, because 5 × 2 = 10 was already marked off by multiple of 2, similarly 5 × 3 = 15 was already marked off by multiple of 3. Therefore, if the current number is p, we can always mark off multiples of p starting at p2, then in increments of p: p2 + p, p2 + 2p, ... Now what should be the terminating loop condition?

It is easy to say that the terminating loop condition is p < n, which is certainly correct but not efficient. Do you still remember Hint #3?

Yes, the terminating loop condition can be p < √n, as all non-primes ≥ √n must have already been marked off. When the loop terminates, all the numbers in the table that are non-marked are prime.

The Sieve of Eratosthenes uses an extra O(n) memory and its runtime complexity is O(n log log n). For the more mathematically inclined readers, you can read more about its algorithm complexity on Wikipedia.

```
public int countPrimes(int n) {
   boolean[] isPrime = new boolean[n];
   for (int i = 2; i < n; i++) {
      isPrime[i] = true;
   }
   // Loop's ending condition is i * i < n instead of i < sqrt(n)
   // to avoid repeatedly calling an expensive function sqrt().
   for (int i = 2; i * i < n; i++) {
      if (!isPrime[i]) continue;
      for (int j = i * i; j < n; j += i) {
         isPrime[j] = false;
      }
   }
   int count = 0;
   for (int i = 2; i < n; i++) {
      if (isPrime[i]) count++;
   }
   return count;
}
```


Isomorphic Strings

Given two strings s and t, determine if they are isomorphic.

Two strings 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.

For example,
Given "egg", "add", return true.

Given "foo", "bar", return false.

Given "paper", "title", return true.

Note:
You may assume both s and t have the same length.

Reverse Linked List

Reverse a singly linked list.

click to show more hints.

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

Course Schedule

There are a total of n courses you have to take, labeled from 0 to n - 1.

Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1]

Given the total number of courses and a list of prerequisite pairs, is it possible for you to finish all courses?

For example:
2, [[1,0]]
There are a total of 2 courses to take. To take course 1 you should have finished course 0. So it is possible.

2, [[1,0],[0,1]]
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.

Note:

The input prerequisites is a graph represented by a list of edges, not adjacency matrices. Read more about how a graph is represented. You may assume that there are no duplicate edges in the input prerequisites.

click to show more hints.

Hints:

This problem is equivalent to finding if a cycle exists in a directed graph. If a cycle exists, no topological ordering exists and therefore it will be impossible to take all courses.
Topological Sort via DFS - A great video tutorial (21 minutes) on Coursera explaining the basic concepts of Topological Sort.
Topological sort could also be done via BFS.

Implement Trie (Prefix Tree)

Implement a trie with insert, search, and startsWith methods.

Note:
You may assume that all inputs are consist of lowercase letters a-z.

Minimum Size Subarray Sum

Given an array of n positive integers and a positive integer s, find the minimal length of a contiguous subarray of which the sum ≥ s. If there isn't one, return 0 instead.

For example, given the array [2,3,1,2,4,3] and s = 7,
the subarray [4,3] has the minimal length under the problem constraint.

click to show more practice.

More practice:

If you have figured out the O(n) solution, try coding another solution of which the time complexity is O(n log n).

Credits:Special thanks to @Freezen for adding this problem and creating all test cases.

Course Schedule II

There are a total of n courses you have to take, labeled from 0 to n - 1.

Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1]

Given the total number of courses and a list of prerequisite pairs, return the ordering of courses you should take to finish all courses.

There may be multiple correct orders, you just need to return one of them. If it is impossible to finish all courses, return an empty array.

For example:
2, [[1,0]]
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]

4, [[1,0],[2,0],[3,1],[3,2]]
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].

Note:

The input prerequisites is a graph represented by a list of edges, not adjacency matrices. Read more about how a graph is represented. You may assume that there are no duplicate edges in the input prerequisites.

click to show more hints.

Hints:

This problem is equivalent to finding the topological order in a directed graph. If a cycle exists, no topological ordering exists and therefore it will be impossible to take all courses.
Topological Sort via DFS - A great video tutorial (21 minutes) on Coursera explaining the basic concepts of Topological Sort.
Topological sort could also be done via BFS.

Add and Search Word - Data structure design

Design a data structure that supports the following two operations:

void addWord(word)
bool search(word)

search(word) can search a literal word or a regular expression string containing only letters a-z or .. A . means it can represent any one letter.

For example:

```
addWord("bad")
addWord("dad")
addWord("mad")
search("pad") -> false
search("bad") -> true
search(".ad") -> true
search("b..") -> true
```

Note:
You may assume that all words are consist of lowercase letters a-z.

click to show hint.

You should be familiar with how a Trie works. If not, please work on this problem: Implement Trie (Prefix Tree) first.

Word Search II

Given a 2D board and a list of words from the dictionary, find all words in the board.

Each word must be constructed from letters of sequentially adjacent cell, where "adjacent" cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once in a word.

For example,
Given words = ["oath","pea","eat","rain"] and board =

```
[
  ['o','a','a','n'],
  ['e','t','a','e'],
  ['i','h','k','r'],
  ['i','f','l','v']
]
```

Return ["eat","oath"].

Note:
You may assume that all inputs are consist of lowercase letters a-z.

click to show hint.

You would need to optimize your backtracking to pass the larger test. Could you stop backtracking earlier?

If the current candidate does not exist in all words' prefix, you could stop backtracking immediately. What kind of data structure could answer such query efficiently? Does a hash table work? Why or why not? How about a Trie? If you would like to learn how to implement a basic trie, please work on this problem: Implement Trie (Prefix Tree) first.

House Robber II

Note: This is an extension of House Robber.

After robbing those houses on that street, the thief has found himself a new place for his thievery so that he will not get too much attention. This time, all houses at this place are arranged in a circle. That means the first house is the neighbor of the last one. Meanwhile, the security system for these houses remain the same as for those in the previous street.

Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight without alerting the police.

Credits:Special thanks to @Freezen for adding this problem and creating all test cases.

Shortest Palindrome

Given a string S, you are allowed to convert it to a palindrome by adding characters in front of it. Find and return the shortest palindrome you can find by performing this transformation.

For example:
Given "aacecaaa", return "aaacecaaa".
Given "abcd", return "dcbabcd".

Credits:Special thanks to @ifanchu for adding this problem and creating all test cases. Thanks to @Freezen for additional test cases.

Kth Largest Element in an Array

Find the kth largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

For example,
Given [3,2,1,5,6,4] and k = 2, return 5.

Note:
You may assume k is always valid, 1 ≤ k ≤ array's length.

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

Combination Sum III

Find all possible combinations of k numbers that add up to a number n, given that only numbers from 1 to 9 can be used and each combination should be a unique set of numbers.

Example 1:
Input: k = 3, n = 7
Output:

[[1,2,4]]

Example 2:
Input: k = 3, n = 9
Output:

[[1,2,6], [1,3,5], [2,3,4]]

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

Contains Duplicate

Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

The Skyline Problem

A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from a distance. Now suppose you are given the locations and height of all the buildings as shown on a cityscape photo (Figure A), write a program to output the skyline formed by these buildings collectively (Figure B).

The geometric information of each building is represented by a triplet of integers [Li, Ri, Hi], where Li and Ri are the x coordinates of the left and right edge of the ith building, respectively, and Hi is its height. It is guaranteed that $0 \le Li, Ri \le INT_MAX$, $0 < Hi \le INT_MAX$, and $Ri - Li > 0$. You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0.

For instance, the dimensions of all buildings in Figure A are recorded as: [[2 9 10], [3 7 15], [5 12 12], [15 20 10], [19 24 8]] .

The output is a list of "key points" (red dots in Figure B) in the format of [[x1,y1], [x2, y2], [x3, y3], ...] that uniquely defines a skyline. A key point is the left endpoint of a horizontal line segment. Note that the last key point, where the rightmost building ends, is merely used to mark the termination of the skyline, and always has zero height. Also, the ground in between any two adjacent buildings should be considered part of the skyline contour.

For instance, the skyline in Figure B should be represented as:[[2 10], [3 15], [7 12], [12 0], [15 10], [20 8], [24, 0]].

Notes:

- The number of buildings in any input list is guaranteed to be in the range [0, 10000].
- The input list is already sorted in ascending order by the left x position Li.
- The output list must be sorted by the x position.
- 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], ...]

Credits:Special thanks to @stellari for adding this problem, creating these two awesome images and all test cases.

Contains Duplicate II

Given an array of integers and an integer k, find out whether there are two distinct indices i and j in the array such that nums[i] = nums[j] and the absolute difference between i and j is at most k.

Contains Duplicate III

Given an array of integers, find out whether there are two distinct indices i and j in the array such that the absolute difference between nums[i] and nums[j] is at most t and the absolute difference between i and j is at most k.

Maximal Square

Given a 2D binary matrix filled with 0's and 1's, find the largest square containing only 1's and return its area.

For example, given the following matrix:

```
1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0
```

Return 4.

Credits:Special thanks to @Freezen for adding this problem and creating all test cases.

Count Complete Tree Nodes

Given a complete binary tree, count the number of nodes.

Definition of a complete binary tree from Wikipedia:
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.

Rectangle Area

Find the total area covered by two rectilinear rectangles in a 2D plane.
Each rectangle is defined by its bottom left corner and top right corner as shown in the figure.

Assume that the total area is never beyond the maximum possible value of int.

Credits:Special thanks to @mithmatt for adding this problem, creating the above image and all test cases.

Basic Calculator

Implement a basic calculator to evaluate a simple expression string.

The expression string may contain open (and closing parentheses), the plus + or minus sign -, non-negative integers and empty spaces .

You may assume that the given expression is always valid.

Some examples:

"1 + 1" = 2
" 2-1 + 2 " = 3
"(1+(4+5+2)-3)+(6+8)" = 23

Note: Do not use the eval built-in library function.

Implement Stack using Queues

Implement the following operations of a stack using queues.

push(x) -- Push element x onto stack.

pop() -- Removes the element on top of the stack.

top() -- Get the top element.

empty() -- Return whether the stack is empty.

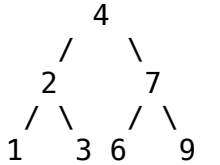
Notes:

You must use only standard operations of a queue -- which means only push to back, peek/pop from front, size, and is empty operations are valid.
Depending on your language, queue may not be supported natively. You may simulate a queue by using a list or deque (double-ended queue), as long as you use only standard operations of a queue.
You may assume that all operations are valid (for example, no pop or top operations will be called on an empty stack).

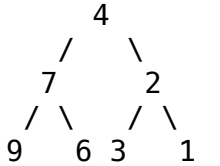
Credits:Special thanks to @jianchao.li.fighter for adding this problem and all test cases.

Invert Binary Tree

Invert a binary tree.



to



Trivia:
This problem was inspired by this original tweet by Max Howell:
Google: 90% of our engineers use the software you wrote (Homebrew), but you can’t invert a binary tree on a whiteboard so fuck off.

Basic Calculator II

Implement a basic calculator to evaluate a simple expression string.

The expression string contains only non-negative integers, +, -, *, / operators and empty spaces . The integer division should truncate toward zero.

You may assume that the given expression is always valid.

Some examples:

"3+2*2" = 7
" 3/2 " = 1
" 3+5 / 2 " = 5

Note: Do not use the eval built-in library function.

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Summary Ranges

Given a sorted integer array without duplicates, return the summary of its ranges.

For example, given [0,1,2,4,5,7], return ["0->2","4->5","7"].

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Majority Element II

Given an integer array of size n, find all elements that appear more than ⌊ n/3 ⌋ times. The algorithm should run in linear time and in O(1) space.

How many majority elements could it possibly have?
Do you have a better hint? Suggest it!

Kth Smallest Element in a BST

Given a binary search tree, write a function kthSmallest to find the kth smallest element in it.

Note:
You may assume k is always valid, 1 ≤ k ≤ BST's total elements.

Follow up:
What if the BST is modified (insert/delete operations) often and you need to find the kth smallest frequently? How would you optimize the kthSmallest routine?

Try to utilize the property of a BST.
What if you could modify the BST node's structure?
The optimal runtime complexity is O(height of BST).

Credits:Special thanks to @ts for adding this problem and creating all test cases.

Power of Two

Given an integer, write a function to determine if it is a power of two.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Implement Queue using Stacks

Implement the following operations of a queue using stacks.

push(x) -- Push element x to the back of queue.

pop() -- Removes the element from in front of queue.

peek() -- Get the front element.

empty() -- Return whether the queue is empty.

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, stack may not be supported natively. You may simulate a stack by using a list or deque (double-ended queue),

as long as you use only standard operations of a stack.
You may assume that all operations are valid (for example, no pop or peek operations will be called on an empty queue).

Number of Digit One

Given an integer n, count the total number of digit 1 appearing in all non-negative integers less than or equal to n.

For example:
Given n = 13,
Return 6, because digit 1 occurred in the following numbers: 1, 10, 11, 12, 13.

Beware of overflow.

Palindrome Linked List

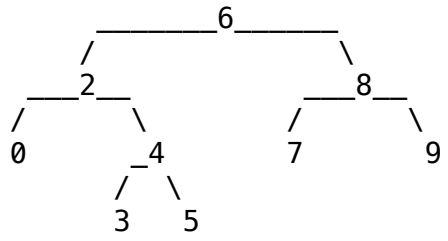
Given a singly linked list, determine if it is a palindrome.

Follow up:
Could you do it in O(n) time and O(1) space?

Lowest Common Ancestor of a Binary Search Tree

Given a binary search tree (BST), find the lowest common ancestor (LCA) of two given nodes in the BST.

According to the definition of LCA on Wikipedia: “The lowest common ancestor is defined between two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself).”

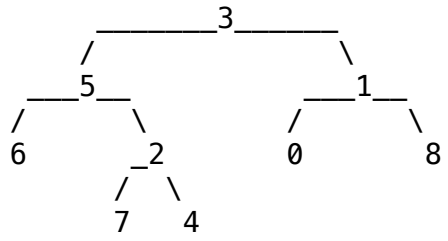


For example, the lowest common ancestor (LCA) of nodes 2 and 8 is 6. Another example is LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

Lowest Common Ancestor of a Binary Tree

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 v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself).”



For example, the lowest common ancestor (LCA) of nodes 5 and 1 is 3. Another example is LCA of nodes 5 and 4 is 5, since a node can be a descendant of itself according to the LCA definition.

Delete Node in a Linked List

Write a function to delete a node (except the tail) in a singly linked list, given only access to that node.

Supposed the linked list is 1 -> 2 -> 3 -> 4 and you are given the third node with value 3, the linked list should become 1 -> 2 -> 4 after calling your function.

Product of Array Except Self

Given an array of n integers where n > 1, nums, return an array output such that output[i] is equal to the product of all the elements of nums except nums[i].

Solve it without division and in O(n).

For example, given [1,2,3,4], return [24,12,8,6].

Follow up:
Could you solve it with constant space complexity? (Note: The output array does not count as extra space for the purpose of space complexity analysis.)

Sliding Window Maximum

Given an array 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.

For example,
Given nums = [1,3,-1,-3,5,3,6,7], and k = 3.

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

Therefore, return the max sliding window as [3,3,5,5,6,7].

Note:
You may assume k is always valid, ie: 1 ≤ k ≤ input array's size for non-empty array.

Follow up:
Could you solve it in linear time?

How about using a data structure such as deque (double-ended queue)?
The queue size need not be the same as the window’s size.
Remove redundant elements and the queue should store only elements that need to be considered.

Search a 2D Matrix II

Write an efficient algorithm that searches for a value in an m x n 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.

For example,

Consider the following 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]
]
```

Given target = 5, return true.
Given target = 20, return false.

Different Ways to Add Parentheses

Given a string of numbers and operators, return all possible results from computing all the different possible ways to group numbers and operators. The valid operators are +, - and *.

Example 1
Input: "2-1-1".
((2-1)-1) = 0
(2-(1-1)) = 2
Output: [0, 2]

Example 2
Input: "2*3-4*5"
(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
Output: [-34, -14, -10, -10, 10]

Credits:Special thanks to @mithmatt for adding this problem and creating all test cases.

Valid Anagram

Given two strings s and t, write a function to determine if t is an anagram of s.

For example,
s = "anagram", t = "nagaram", return true.
s = "rat", t = "car", return false.

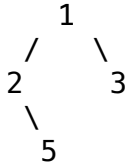
Note:
You may assume the string contains only lowercase alphabets.

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

Binary Tree Paths

Given a binary tree, return all root-to-leaf paths.

For example, given the following binary tree:



All root-to-leaf paths are:
["1->2->5", "1->3"]

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Add Digits

Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.

For example:

Given num = 38, the process is like: 3 + 8 = 11, 1 + 1 = 2. Since 2 has only one digit, return it.

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

A naive implementation of the above process is trivial. Could you come up with other methods?
What are all the possible results?
How do they occur, periodically or randomly?
You may find this Wikipedia article useful.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Single Number III

Given an array of numbers 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.

For example:

Given nums = [1, 2, 1, 3, 2, 5], return [3, 5].

Note:

The order of the result is not important. So in the above example, [5, 3] is also correct.
Your algorithm should run in linear runtime complexity. Could you implement it using only constant space complexity?

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Ugly Number

Write a program to check whether a given number is an ugly number.

Ugly numbers are positive numbers whose prime factors only include 2, 3, 5. For example, 6, 8 are ugly while 14 is not ugly since it includes another prime factor 7.

Note that 1 is typically treated as an ugly number.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Ugly Number II

Write a program to find the n-th ugly number.

Ugly numbers are positive numbers whose prime factors only include 2, 3, 5. For example, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12 is the sequence of the first 10 ugly numbers.

Note that 1 is typically treated as an ugly number, and n does not exceed 1690.

The naive approach is to call isUgly for every number until you reach the nth one. Most numbers are not ugly. Try to focus your effort on generating only the ugly ones.

An ugly number must be multiplied by either 2, 3, or 5 from a

smaller ugly number.
  The key is how to maintain the order of the ugly numbers. Try a similar approach of merging from three sorted lists: L1, L2, and L3.
  Assume you have Uk, the kth ugly number. Then Uk+1 must be Min(L1 * 2, L2 * 3, L3 * 5).

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Missing Number

Given an array containing n distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing from the array.

For example,
Given nums = [0, 1, 3] return 2.

Note:
Your algorithm should run in linear runtime complexity. Could you implement it using only constant extra space complexity?

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Integer to English Words

Convert a non-negative integer to its english words representation. Given input is guaranteed to be less than 231 - 1.

For example,

123 -> "One Hundred Twenty Three"
12345 -> "Twelve Thousand Three Hundred Forty Five"
1234567 -> "One Million Two Hundred Thirty Four Thousand Five Hundred Sixty Seven"

  Did you see a pattern in dividing the number into chunk of words?

For example, 123 and 123000.

Group the number by thousands (3 digits). You can write a helper function that takes a number less than 1000 and convert just that chunk to words.

There are many edge cases. What are some good test cases? Does your code work with input such as 0? Or 1000010? (middle chunk is zero and should not be printed out)

H-Index

Given an array of citations (each citation is a non-negative integer) of a researcher, write a function to compute the researcher's h-index.

According to the definition of h-index on Wikipedia: "A scientist has index h if h of his/her N papers have at least h citations each, and the other N − h papers have no more than h citations each."

For example, given citations = [3, 0, 6, 1, 5], which 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, his h-index is 3.

Note: If there are several possible values for h, the maximum one is taken as the h-index.

An easy approach is to sort the array first.
What are the possible values of h-index?
A faster approach is to use extra space.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

H-Index II

Follow up for H-Index: What if the citations array is sorted in ascending order? Could you optimize your algorithm?

Expected runtime complexity is in O(log n) and the input is sorted.

First Bad Version

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 will return whether version is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Perfect Squares

Given a positive integer n, find the least number of perfect square numbers (for example, 1, 4, 9, 16, ...) which sum to n.

For example, given n = 12, return 3 because 12 = 4 + 4 + 4; given n = 13, return 2 because 13 = 4 + 9.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Expression Add Operators

Given a string that contains only digits 0-9 and a target value,

return all possibilities to add binary operators (not unary) +, -, or * between the digits so they evaluate to the target value.

Examples:
"123", 6 -> ["1+2+3", "1*2*3"]
"232", 8 -> ["2*3+2", "2+3*2"]
"105", 5 -> ["1*0+5","10-5"]
"00", 0 -> ["0+0", "0-0", "0*0"]
"3456237490", 9191 -> []

Credits:Special thanks to @davidtan1890 for adding this problem and creating all test cases.

Move Zeroes

Given an array nums, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

For example, given nums = [0, 1, 0, 3, 12], after calling your function, nums should be [1, 3, 12, 0, 0].

Note:

You must do this in-place without making a copy of the array.
Minimize the total number of operations.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Peeking Iterator

Given an Iterator class interface with methods: next() and hasNext(), design and implement a PeekingIterator that support the peek() operation -- it essentially peek() at the element that will be returned by the next call to next().

Here is an example. Assume that the iterator is initialized to the beginning of the list: [1, 2, 3].

Call next() gets you 1, the first element in the list.

Now you call peek() and it returns 2, the next element. Calling

next() after that still return 2.

You call next() the final time and it returns 3, the last element. Calling hasNext() after that should return false.

Think of "looking ahead". You want to cache the next element.
Is one variable sufficient? Why or why not?
Test your design with call order of peek() before next() vs next() before peek().
For a clean implementation, check out Google's guava library source code.

Follow up: How would you extend your design to be generic and work with all types, not just integer?

Credits:Special thanks to @porker2008 for adding this problem and creating all test cases.

Find the Duplicate Number

Given an array nums containing n + 1 integers where each integer is between 1 and n (inclusive), prove that at least one duplicate number must exist. Assume that there is only one duplicate number, find the duplicate one.

Note:

You must not modify the array (assume the array is read only).
You must use only constant, O(1) extra space.
Your runtime complexity should be less than O(n2).
There is only one duplicate number in the array, but it could be repeated more than once.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Game of Life

According to the 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."

Given a board with m by n cells, each cell has an initial state live (1) or dead (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.

Write a function to compute the next state (after one update) of the board given its current state.

Follow up:

Could you solve it in-place? Remember that the board needs to be updated at the same time: 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 the border of the array. How would you address these problems?

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Word Pattern

Given a pattern and a string str, find if str 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 str.

Examples:

pattern = "abba", str = "dog cat cat dog" should return true.

pattern = "abba", str = "dog cat cat fish" should return false.
pattern = "aaaa", str = "dog cat cat dog" should return false.
pattern = "abba", str = "dog dog dog dog" should return false.

Notes:
You may assume pattern contains only lowercase letters, and str contains lowercase letters separated by a single space.

Credits:Special thanks to @minglotus6 for adding this problem and creating all test cases.

Nim Game

You are playing the following Nim Game with your friend: There is a heap of stones on the table, each time one of you take turns to remove 1 to 3 stones. The one who removes the last stone will be the winner. You will take the first turn to remove the stones.

Both of you are very clever and have optimal strategies for the game. Write a function to determine whether you can win the game given the number of stones in the heap.

For example, if there are 4 stones in the heap, then you will never win the game: no matter 1, 2, or 3 stones you remove, the last stone will always be removed by your friend.

If there are 5 stones in the heap, could you figure out a way to remove the stones such that you will always be the winner?

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Find Median from Data Stream

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 value.

Examples:
[2,3,4] , the median is 3
[2,3], the median is (2 + 3) / 2 = 2.5

Design a data structure that supports the following two operations:

void addNum(int num) - Add a integer number from the data stream to the data structure.
double findMedian() - Return the median of all elements so far.

For example:

addNum(1)
addNum(2)
findMedian() -> 1.5
addNum(3)
findMedian() -> 2

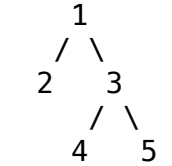
Credits:Special thanks to @Louis1992 for adding this problem and creating all test cases.

Serialize and Deserialize Binary Tree

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.

For example, you may serialize the following tree



as "[1,2,3,null,null,4,5]", just the same as how LeetCode OJ serializes a binary tree. You do not necessarily need to follow this format, so please be creative and come up with different approaches yourself.

Note: Do not use class member/global/static variables to store states. Your serialize and deserialize algorithms should be stateless.

Credits:Special thanks to @Louis1992 for adding this problem and creating all test cases.

Bulls and Cows

You are playing the following Bulls and Cows game with your friend: You write down a number and ask your friend to guess what the number is. Each time your friend makes a guess, you provide a hint that indicates how many digits in said guess match your secret number exactly in both digit and position (called "bulls") and how many digits match the secret number but locate in the wrong position (called "cows"). Your friend will use successive guesses and hints to eventually derive the secret number.

For example:

Secret number: "1807"
Friend's guess: "7810"

Hint: 1 bull and 3 cows. (The bull is 8, the cows are 0, 1 and 7.)

Write a function to return a hint according to the secret number and friend's guess, use A to indicate the bulls and B to indicate the cows. In the above example, your function should return "1A3B".

Please note that both secret number and friend's guess may contain duplicate digits, for example:

Secret number: "1123"
Friend's guess: "0111"

In this case, the 1st 1 in friend's guess is a bull, the 2nd or 3rd 1 is a cow, and your function should return "1A1B".

You may assume that the secret number and your friend's guess only contain digits, and their lengths are always equal.

Credits:Special thanks to @jeantimex for adding this problem and creating all test cases.

Longest Increasing Subsequence

Given an unsorted array of integers, find the length of longest increasing subsequence.

For example,
Given [10, 9, 2, 5, 3, 7, 101, 18],
The longest increasing subsequence is [2, 3, 7, 101], therefore the length is 4. Note that there may be more than one LIS combination, it is only necessary for you to return the length.

Your algorithm should run in O(n2) complexity.

Follow up: Could you improve it to O(n log n) time complexity?

Credits:Special thanks to @pbrother for adding this problem and creating all test cases.

Remove Invalid Parentheses

Remove the minimum number of invalid parentheses in order to make the input string valid. Return all possible results.

Note: The input string may contain letters other than the parentheses (and).

Examples:

"()())()" -> ["()()()", "(())()"]
"(a())()" -> ["(a)()()", "(a())()"]
")(" -> [""]

Credits:Special thanks to @hpplayer for adding this problem and creating all test cases.

Range Sum Query - Immutable

Given an integer array nums, find the sum of the elements between indices i and j (i ≤ j), inclusive.

Example:

```
Given nums = [-2, 0, 3, -5, 2, -1]

sumRange(0, 2) -> 1
sumRange(2, 5) -> -1
sumRange(0, 5) -> -3
```

Note:

You may assume that the array does not change.
There are many calls to sumRange function.

Range Sum Query 2D - Immutable

Given a 2D matrix matrix, find the sum of the elements inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

The above rectangle (with the red border) is defined by (row1, col1) = (2, 1) and (row2, col2) = (4, 3), which contains sum = 8.

Example:

```
Given matrix = [
  [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]
]

sumRegion(2, 1, 4, 3) -> 8
sumRegion(1, 1, 2, 2) -> 11
sumRegion(1, 2, 2, 4) -> 12
```

Note:

You may assume that the matrix does not change.
There are many calls to sumRegion function.
You may assume that row1 ≤ row2 and col1 ≤ col2.

Additive Number

Additive number is a string whose digits can form 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.

For example:
"112358" is an additive number because 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
"199100199" is also an additive number, the additive sequence is: 1, 99, 100, 199.
1 + 99 = 100, 99 + 100 = 199

Note: Numbers in the additive sequence cannot have leading zeros, so sequence 1, 2, 03 or 1, 02, 3 is invalid.

Given a string containing only digits '0'-'9', write a function to determine if it's an additive number.

Follow up:
How would you handle overflow for very large input integers?

Credits:Special thanks to @jeantimex for adding this problem and creating all test cases.

Range Sum Query - Mutable

Given an integer array nums, find the sum of the elements between indices i and j (i ≤ j), inclusive.

The update(i, val) function modifies nums by updating the element at index i to val.

Example:

Given nums = [1, 3, 5]

sumRange(0, 2) -> 9
update(1, 2)
sumRange(0, 2) -> 8

Note:

The array is only modifiable by the update function.
You may assume the number of calls to update and sumRange function is distributed evenly.

Minimum Height Trees

For a undirected graph with tree characteristics, we can choose any node as the root. The result graph is then a rooted tree. Among all possible rooted trees, those with minimum height are called minimum height trees (MHTs).

Given such a graph, write a function to find all the MHTs and return a list of their root labels.

Format

The graph contains n nodes which are labeled from 0 to n - 1.

You will be given the number n and a list of undirected edges (each edge is a pair of labels).

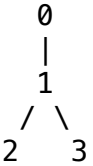
You can assume that no duplicate edges will appear in edges. Since all edges are

undirected, [0, 1] is the same as [1, 0] and thus will not appear together in

edges.

Example 1:

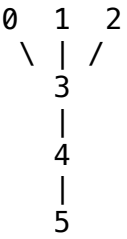
Given n = 4, edges = [[1, 0], [1, 2], [1, 3]]



return [1]

Example 2:

Given n = 6, edges = [[0, 3], [1, 3], [2, 3], [4, 3], [5, 4]]



return [3, 4]

How many MHTs can a graph have at most?

Note:

(1) According to the definition of tree on Wikipedia: “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.”

(2) The height of a rooted tree is the number of edges on the longest downward path between the root and a leaf.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Burst Balloons

Given n balloons, indexed from 0 to n-1. Each balloon is painted with a number on it represented by array nums.

You are asked to burst all the balloons. If the you burst balloon i you will get nums[left] * nums[i] * nums[right] coins. Here left and right are adjacent indices of i. After the burst, the left and right then becomes adjacent.

Find the maximum coins you can collect by bursting the balloons wisely.

Note:
(1) You may imagine nums[-1] = nums[n] = 1. They are not real therefore you can not burst them.
(2) 0 ≤ n ≤ 500, 0 ≤ nums[i] ≤ 100

Example:

Given [3, 1, 5, 8]

Return 167

```
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
```

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Super Ugly Number

Write a program to find the nth super ugly number.

Super ugly numbers are positive numbers whose all prime factors are in the given prime list
primes of size k. For example, [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] of size 4.

Note:
(1) 1 is a super ugly number for any given primes.
(2) The given numbers in primes are in ascending order.
(3) 0 < k ≤ 100, 0 < n ≤ 106, 0 < primes[i] < 1000.
(4) The nth super ugly number is guaranteed to fit in a 32-bit signed integer.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Count of Smaller Numbers After Self

You are given an integer array nums and you have to return a new counts array.
The counts array has the property where counts[i] is
the number of smaller elements to the right of nums[i].

Example:

Given nums = [5, 2, 6, 1]

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.

Return the array [2, 1, 1, 0].

Remove Duplicate Letters

Given a string which contains only lowercase letters, remove duplicate letters so that every letter appear once and only once. You must make sure your result is the smallest in lexicographical order among all possible results.

Example:

Given "bcabc"
Return "abc"

Given "cbacdcbc"
Return "acdb"

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Maximum Product of Word Lengths

Given a string array words, find the maximum value of length(word[i]) * length(word[j]) where the two words do not share common letters.
You may assume that each word will contain only lower case letters.
If no such two words exist, return 0.

Example 1:

Given ["abcw", "baz", "foo", "bar", "xtfn", "abcdef"]
Return 16
The two words can be "abcw", "xtfn".

Example 2:

Given ["a", "ab", "abc", "d", "cd", "bcd", "abcd"]
Return 4
The two words can be "ab", "cd".

Example 3:

Given ["a", "aa", "aaa", "aaaa"]
Return 0
No such pair of words.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Bulb Switcher

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.

Find how many bulbs are on after n rounds.

Example:

Given n = 3.
At first, the three bulbs are [off, off, off].
After first round, the three bulbs are [on, on, on].
After second round, the three bulbs are [on, off, on].
After third round, the three bulbs are [on, off, off].
So you should return 1, because there is only one bulb is on.

Create Maximum Number

Given two arrays of length m and n with digits 0-9 representing two numbers.
Create the maximum number of length k <= m + n from digits of the two. The relative order of the digits
from the same array must be preserved. Return an array of the k digits. You should try to optimize your time and space complexity.

Example 1:

nums1 = [3, 4, 6, 5]
nums2 = [9, 1, 2, 5, 8, 3]
k = 5
return [9, 8, 6, 5, 3]

Example 2:

nums1 = [6, 7]
nums2 = [6, 0, 4]
k = 5
return [6, 7, 6, 0, 4]

Example 3:

nums1 = [3, 9]
nums2 = [8, 9]
k = 3
return [9, 8, 9]

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Coin Change

You are given coins of different denominations and a total amount of money amount. Write a function to compute 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.

Example 1:
coins = [1, 2, 5], amount = 11
return 3 (11 = 5 + 5 + 1)

Example 2:
coins = [2], amount = 3
return -1.

Note:
You may assume that you have an infinite number of each kind of coin.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Wiggle Sort II

Given an unsorted array nums, reorder it such that nums[0] < nums[1] > nums[2] < nums[3]....

Example:
(1) Given nums = [1, 5, 1, 1, 6, 4], one possible answer is [1, 4, 1, 5, 1, 6].
(2) Given nums = [1, 3, 2, 2, 3, 1], one possible answer is [2, 3, 1, 3, 1, 2].

Note:
You may assume all input has valid answer.

Follow Up:
Can you do it in O(n) time and/or in-place with O(1) extra space?

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Power of Three

Given an integer, write a function to determine if it is a power of three.

Follow up:
Could you do it without using any loop / recursion?

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Count of Range Sum

Given an integer array nums, 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 (i ≤ j), inclusive.

Note:
A naive algorithm of O(n2) is trivial. You MUST do better than that.

Example:
Given nums = [-2, 5, -1], lower = -2, upper = 2,
Return 3.
The three ranges are : [0, 0], [2, 2], [0, 2] and their respective sums are: -2, -1, 2.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Odd Even Linked List

Given a singly linked list, group all odd nodes together followed by the even nodes. Please note here we are talking about the node number and not the value in the nodes.

You should try to do it in place. The program should run in O(1) space complexity and O(nodes) time complexity.

Example:
Given 1->2->3->4->5->NULL,
return 1->3->5->2->4->NULL.

Note:
The relative order inside both the even and odd groups should remain as it was in the input.
The first node is considered odd, the second node even and so on ...

Credits:Special thanks to @DjangoUnchained for adding this problem and creating all test cases.

Longest Increasing Path in a Matrix

Given an integer matrix, find the length of the longest increasing path.

From each cell, you can either move to four directions: left, right, up or down. You may NOT move diagonally or move outside of the boundary (i.e. wrap-around is not allowed).

Example 1:

```
nums = [
  [9,9,4],
  [6,6,8],
  [2,1,1]
]
```

Return 4

The longest increasing path is [1, 2, 6, 9].

Example 2:

```
nums = [
  [3,4,5],
  [3,2,6],
  [2,2,1]
]
```

Return 4

The longest increasing path is [3, 4, 5, 6]. Moving diagonally is not allowed.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Patching Array

Given a sorted positive integer array nums and an integer n, add/patch elements to the array such that any number in 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:
nums = [1, 3], n = 6
Return 1.

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:
nums = [1, 5, 10], n = 20
Return 2.
The two patches can be [2, 4].

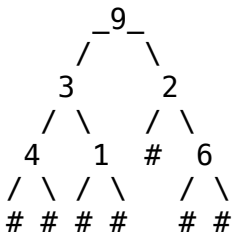
Example 3:
nums = [1, 2, 2], n = 5
Return 0.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Verify Preorder Serialization of a Binary Tree

One way to serialize a binary tree is to use pre-order 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, verify whether it is a correct preorder traversal serialization of a binary tree. Find an algorithm without reconstructing the tree.

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".

Example 1:
"9,3,4,#,#,1,#,#,2,#,6,#,#"
Return true
Example 2:
"1,#"
Return false
Example 3:
"9,#,#,1"
Return false

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Reconstruct Itinerary

Given a list of airline tickets represented by pairs of departure and arrival airports [from, to], reconstruct the itinerary in order. All of the tickets belong to a man who departs from JFK. Thus, the itinerary must begin with JFK.

Note:

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"].
All airports are represented by three capital letters (IATA code).
You may assume all tickets form at least one valid itinerary.

Example 1:
tickets = [["MUC", "LHR"], ["JFK", "MUC"], ["SFO", "SJC"], ["LHR", "SFO"]]
Return ["JFK", "MUC", "LHR", "SFO", "SJC"].

Example 2:
tickets = [["JFK","SFO"],["JFK","ATL"],["SFO","ATL"], ["ATL","JFK"],["ATL","SFO"]]
Return ["JFK","ATL","JFK","SFO","ATL","SFO"].
Another possible reconstruction is ["JFK","SFO","ATL","JFK","ATL","SFO"]. But it is larger in lexical order.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Increasing Triplet Subsequence

Given an unsorted array return whether an increasing subsequence of length 3 exists or not in the array.

Formally the function should:
Return true if there exists i, j, k
such that arr[i] < arr[j] < arr[k] given 0 ≤ i < j < k ≤ n-1
else return false.

Your algorithm should run in O(n) time complexity and O(1) space complexity.

Examples:
Given [1, 2, 3, 4, 5],
return true.

Given [5, 4, 3, 2, 1],
return false.

Credits:Special thanks to @DjangoUnchained for adding this problem and creating all test cases.

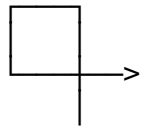
Self Crossing

You are given an array x of n positive numbers. You start at point (0,0) and moves x[0] metres to the north, then x[1] metres to the west,
x[2] metres to the south,
x[3] metres to the east and so on. In other words, after each move your direction changes
counter-clockwise.

Write a one-pass algorithm with O(1) extra space to determine, if your path crosses itself, or not.

Example 1:

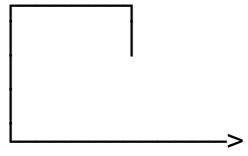
Given x = [2, 1, 1, 2],



Return true (self crossing)

Example 2:

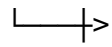
Given x = [1, 2, 3, 4],



Return false (not self crossing)

Example 3:

Given x = [1, 1, 1, 1],



Return true (self crossing)

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Palindrome Pairs

Given a list of unique words, find all pairs of distinct indices (i, j) in the given list, so that the concatenation of the two words, i.e. words[i] + words[j] is a palindrome.

Example 1:
Given words = ["bat", "tab", "cat"]
Return [[0, 1], [1, 0]]
The palindromes are ["battab", "tabbat"]

Example 2:
Given words = ["abcd", "dcba", "lls", "s", "sssll"]
Return [[0, 1], [1, 0], [3, 2], [2, 4]]
The palindromes are ["dcbaabcd", "abcddcba", "slls", "llssssll"]

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

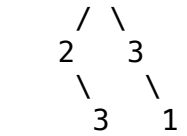
House Robber III

The thief has found himself a new place for his thievery again. There is only one entrance to this area, called the "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 forms a binary tree". It will automatically contact the police if two directly-linked houses were broken into on the same night.

Determine the maximum amount of money the thief can rob tonight without alerting the police.

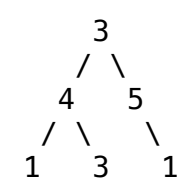
Example 1:

3



Maximum amount of money the thief can rob = 3 + 3 + 1 = 7.

Example 2:



Maximum amount of money the thief can rob = 4 + 5 = 9.

Credits:Special thanks to @dietpepsi for adding this problem and creating all test cases.

Counting Bits

Given a non negative integer number num. For every numbers i in the range 0 ≤ i ≤ num calculate the number of 1's in their binary representation and return them as an array.

Example:
For num = 5 you should return [0,1,1,2,1,2].

Follow up:

It is very easy to come up with a solution with run time O(n*sizeof(integer)). But can you do it in linear time O(n) / possibly in a single pass?
Space complexity should be O(n).
Can you do it like a boss? Do it without using any builtin function like __builtin_popcount in c++ or in any other language.

  You should make use of what you have produced already.
  Divide the numbers in ranges like [2-3], [4-7], [8-15] and so on. And try to generate new range from previous.
  Or does the odd/even status of the number help you in calculating the number of 1s?

Credits:Special thanks to @ syedee for adding this problem and

creating all test cases.

Flatten Nested List Iterator

Given a nested list of integers, implement an iterator to flatten it.

Each element is either an integer, or a list -- whose elements may also be integers or other lists.

Example 1:
Given the list [[1,1],2,[1,1]],

By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,1,2,1,1].

Example 2:
Given the list [1,[4,[6]]],

By calling next repeatedly until hasNext returns false, the order of elements returned by next should be: [1,4,6].

Power of Four

Given an integer (signed 32 bits), write a function to check whether it is a power of 4.

Example:
Given num = 16, return true.
Given num = 5, return false.

Follow up: Could you solve it without loops/recursion?

Credits:Special thanks to @yukuairoy for adding this problem and creating all test cases.

Integer Break

Given a positive integer n, break it into the sum of at least two positive integers and maximize the product of those integers. Return the maximum product you can get.

For example, given n = 2, return 1 (2 = 1 + 1); given n = 10, return 36 (10 = 3 + 3 + 4).

Note: You may assume that n is not less than 2 and not larger than 58.

There is a simple O(n) solution to this problem.
You may check the breaking results of n ranging from 7 to 10 to discover the regularities.

Credits:Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Reverse String

Write a function that takes a string as input and returns the string reversed.

Example:
Given s = "hello", return "olleh".

Reverse Vowels of a String

Write a function that takes a string as input and reverse only the vowels of a string.

Example 1:
Given s = "hello", return "holle".

Example 2:
Given s = "leetcode", return "leotcede".

Note:
The vowels does not include the letter "y".

Top K Frequent Elements

Given a non-empty array of integers, return the k most frequent elements.

For example,
Given [1,1,1,2,2,3] and k = 2, return [1,2].

Note:

You may assume k is always valid, 1 ≤ k ≤ number of unique elements.
Your algorithm's time complexity must be better than O(n log n), where n is the array's size.

Intersection of Two Arrays

Given two arrays, write a function to compute their intersection.

Example:
Given nums1 = [1, 2, 2, 1], nums2 = [2, 2], return [2].

Note:

Each element in the result must be unique.
The result can be in any order.

Intersection of Two Arrays II

Given two arrays, write a function to compute their intersection.

Example:
Given nums1 = [1, 2, 2, 1], nums2 = [2, 2], return [2, 2].

Note:

Each element in the result should appear as many times as it shows in both arrays.
The result can be in any order.

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?

Data Stream as Disjoint Intervals

Given a data stream input of non-negative integers a1, a2, ..., an, ..., summarize the numbers seen so far as a list of disjoint intervals.

For example, suppose the integers from the data stream are 1, 3, 7, 2, 6, ..., then the summary will be:

[1, 1]
[1, 1], [3, 3]
[1, 1], [3, 3], [7, 7]
[1, 3], [7, 7]
[1, 3], [6, 7]

Follow up:
What if there are lots of merges and the number of disjoint intervals are small compared to the data stream's size?

Credits:Special thanks to @yunhong for adding this problem and creating most of the test cases.

Russian Doll Envelopes

You have a number of envelopes with widths and heights given as a pair of integers (w, h). One envelope can fit into another if and only if both the width and height of one envelope is greater than the width and height of the other envelope.

What is the maximum number of envelopes can you Russian doll? (put one inside other)

Example:

Given envelopes = [[5,4],[6,4],[6,7],[2,3]], the maximum number of envelopes you can Russian doll is 3 ([2,3] => [5,4] => [6,7]).

Design Twitter

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. Your design should support the following methods:

postTweet(userId, tweetId): Compose a new tweet.
getNewsFeed(userId): Retrieve 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 herself. Tweets must be ordered from most recent to least recent.
follow(followerId, followeeId): Follower follows a followee.
unfollow(followerId, followeeId): Follower unfollows a followee.

Example:

```
Twitter twitter = new Twitter();

// User 1 posts a new tweet (id = 5).
twitter.postTweet(1, 5);

// User 1's news feed should return a list with 1 tweet id -> [5].
twitter.getNewsFeed(1);

// User 1 follows user 2.
twitter.follow(1, 2);

// User 2 posts a new tweet (id = 6).
twitter.postTweet(2, 6);

// 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.getNewsFeed(1);

// User 1 unfollows user 2.
twitter.unfollow(1, 2);

// User 1's news feed should return a list with 1 tweet id -> [5],
// since user 1 is no longer following user 2.
twitter.getNewsFeed(1);
```


Max Sum of Rectangle No Larger Than K

Given a non-empty 2D matrix matrix and an integer k, find the max sum of a rectangle in the matrix such that its sum is no larger than k.

Example:
Given matrix = [
  [1, 0, 1],
  [0, -2, 3]
]
k = 2

The answer is 2. Because the sum of rectangle [[0, 1], [-2, 3]] is 2 and 2 is the max number no larger than k (k = 2).

Note:

The rectangle inside the matrix must have an area > 0.
What if the number of rows is much larger than the number of columns?

Credits:Special thanks to @fujiaozhu for adding this problem and creating all test cases.

Water and Jug Problem

You are given two jugs with capacities x and y litres. There is an infinite amount of water supply available.
You need to determine whether it is possible to measure exactly z litres using these two jugs.

If z liters of water is measurable, you must have z liters of water contained within one or both buckets by the end.

Operations allowed:

Fill any of the jugs completely 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: (From the famous "Die Hard" example)

Input: x = 3, y = 5, z = 4
Output: True

Example 2:

Input: x = 2, y = 6, z = 5
Output: False

Credits:Special thanks to @vinod23 for adding this problem and creating all test cases.

Valid Perfect Square

Given a positive integer num, write a function which returns True if num is a perfect square else False.

Note: Do not use any built-in library function such as sqrt.

Example 1:

Input: 16
Returns: True

Example 2:

Input: 14
Returns: False

Credits:Special thanks to @elmirap for adding this problem and creating all test cases.

Largest Divisible Subset

Given a set of distinct positive integers, find the largest subset such that every pair (Si, Sj) of elements in this subset satisfies: Si % Sj = 0 or Sj % Si = 0.

If there are multiple solutions, return any subset is fine.

Example 1:

nums: [1,2,3]

Result: [1,2] (of course, [1,3] will also be ok)

Example 2:

nums: [1,2,4,8]

Result: [1,2,4,8]

Credits:Special thanks to @Stomach_ache for adding this problem and creating all test cases.

Sum of Two Integers

Calculate the sum of two integers a and b, but you are not allowed to use the operator + and -.

Example:
Given a = 1 and b = 2, return 3.

Credits:Special thanks to @fujiaozhu for adding this problem and creating all test cases.

Super Pow

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.

Example1:

a = 2

b = [3]

Result: 8

Example2:

a = 2
b = [1,0]

Result: 1024

Credits:Special thanks to @Stomach_ache for adding this problem and creating all test cases.

Find K Pairs with Smallest Sums

You are given two integer arrays nums1 and nums2 sorted in ascending 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.

Find the k pairs (u1,v1),(u2,v2) ...(uk,vk) with the smallest sums.

Example 1:

Given nums1 = [1,7,11], nums2 = [2,4,6], k = 3

Return: [1,2],[1,4],[1,6]

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:

Given nums1 = [1,1,2], nums2 = [1,2,3], k = 2

Return: [1,1],[1,1]

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:

Given nums1 = [1,2], nums2 = [3], k = 3

Return: [1,3],[2,3]

All possible pairs are returned from the sequence:
[1,3],[2,3]

Credits:Special thanks to @elmirap and @StefanPochmann for adding this problem and creating all test cases.

Guess Number Higher or Lower

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'll tell you whether the number is higher or lower.

You call a pre-defined API guess(int num) which returns 3 possible results (-1, 1, or 0):

```
-1 : My number is lower
 1 : My number is higher
 0 : Congrats! You got it!
```

Example:

n = 10, I pick 6.

Return 6.

Guess Number Higher or Lower II

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'll tell you whether the number I picked is higher or lower.

However, when you guess a particular number x, and you guess wrong, you pay $x. You win the game when you guess the number I picked.

Example:

n = 10, I pick 8.

First round: You guess 5, I tell you that it's higher. You pay $5.
Second round: You guess 7, I tell you that it's higher. You pay $7.
Third round: You guess 9, I tell you that it's lower. You pay $9.

Game over. 8 is the number I picked.

You end up paying $5 + $7 + $9 = $21.

Given a particular n ≥ 1, find out how much money you need to have to guarantee a win.

The best strategy to play the game is to minimize the maximum loss you could possibly face. Another strategy is to minimize the expected loss. Here, we are interested in the first scenario.
Take a small example (n = 3). What do you end up paying in the worst case?
Check out this article if you're still stuck.
The purely recursive implementation of minimax would be worthless for even a small n. You MUST use dynamic programming.
As a follow-up, how would you modify your code to solve the problem of minimizing the expected loss, instead of the worst-case loss?

Credits:Special thanks to @agave and @StefanPochmann for adding this problem and creating all test cases.

Wiggle Subsequence

A sequence of numbers is called a wiggle sequence if 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 fewer than two elements is trivially a wiggle sequence.

For example, [1,7,4,9,2,5] is a wiggle sequence because the differences (6,-3,5,-7,3) are alternately positive and negative. In contrast, [1,4,7,2,5] and [1,7,4,5,5] are not wiggle sequences, the first because its first two differences are positive and the second because its last difference is zero.

Given a sequence of integers, return the length of the longest subsequence that is a wiggle sequence. A subsequence is obtained by deleting some number of elements (eventually, also zero) from the original sequence, leaving the remaining elements in their original

order.

Examples:

Input: [1,7,4,9,2,5]
Output: 6
The entire sequence is a wiggle sequence.

Input: [1,17,5,10,13,15,10,5,16,8]
Output: 7
There are several subsequences that achieve this length. One is [1,17,10,13,10,16,8].

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

Follow up:
Can you do it in O(n) time?

Credits:Special thanks to @agave and @StefanPochmann for adding this problem and creating all test cases.

Combination Sum IV

Given an integer array with all positive numbers and no duplicates, find the number of possible combinations that add up to a positive integer target.

Example:

nums = [1, 2, 3]
target = 4

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.

Therefore the output is 7.

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?

Credits:Special thanks to @pbrother for adding this problem and creating all test cases.

Kth Smallest Element in a Sorted Matrix

Given a n x n matrix where each of the rows and columns are sorted in ascending order, find the kth smallest element in the matrix.

Note that it is the kth smallest element in the sorted order, not the kth distinct element.

Example:

```
matrix = [
   [ 1,  5,  9],
   [10, 11, 13],
   [12, 13, 15]
],
k = 8,

return 13.
```

Note:
You may assume k is always valid, 1 ≤ k ≤ n2.

Insert Delete GetRandom O(1)

Design a data structure that supports all following operations in average O(1) time.

insert(val): Inserts an item val to the set if not already present.
remove(val): Removes an item val from the set if present.
getRandom: Returns a random element from current set of elements. Each element must have the same probability of being returned.

Example:

```
// Init an empty set.
RandomizedSet randomSet = new RandomizedSet();
```

```
// Inserts 1 to the set. Returns true as 1 was inserted
successfully.
randomSet.insert(1);

// Returns false as 2 does not exist in the set.
randomSet.remove(2);

// Inserts 2 to the set, returns true. Set now contains [1,2].
randomSet.insert(2);

// getRandom should return either 1 or 2 randomly.
randomSet.getRandom();

// Removes 1 from the set, returns true. Set now contains [2].
randomSet.remove(1);

// 2 was already in the set, so return false.
randomSet.insert(2);

// Since 2 is the only number in the set, getRandom always return 2.
randomSet.getRandom();
```


Insert Delete GetRandom O(1) - Duplicates allowed

Design a data structure that supports all following operations in average O(1) time.
Note: Duplicate elements are allowed.

insert(val): Inserts an item val to the collection.
remove(val): Removes an item val from the collection if present.
getRandom: Returns a random element from current collection of elements. The probability of each element being returned is linearly related to the number of same value the collection contains.

Example:

```
// Init an empty collection.
RandomizedCollection collection = new RandomizedCollection();

// Inserts 1 to the collection. Returns true as the collection did
not contain 1.
collection.insert(1);

// Inserts another 1 to the collection. Returns false as the
collection contained 1. Collection now contains [1,1].
collection.insert(1);
```

```
// Inserts 2 to the collection, returns true. Collection now
contains [1,1,2].
collection.insert(2);

// getRandom should return 1 with the probability 2/3, and returns 2
with the probability 1/3.
collection.getRandom();

// Removes 1 from the collection, returns true. Collection now
contains [1,2].
collection.remove(1);

// getRandom should return 1 and 2 both equally likely.
collection.getRandom();
```


Linked List Random Node

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.

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?

Example:

```
// Init a singly linked list [1,2,3].
ListNode head = new ListNode(1);
head.next = new ListNode(2);
head.next.next = new ListNode(3);
Solution solution = new Solution(head);

// getRandom() should return either 1, 2, or 3 randomly. Each
element should have equal probability of returning.
solution.getRandom();
```


Ransom Note

Given an arbitrary ransom note string and another string containing letters from all the magazines, write a function that will return true if the ransom
note can be constructed from the magazines ; otherwise, it will return false.

Each letter in the magazine string can only be used once in your ransom note.

Note:
You may assume that both strings contain only lowercase letters.

canConstruct("a", "b") -> false
canConstruct("aa", "ab") -> false
canConstruct("aa", "aab") -> true

Shuffle an Array

Shuffle a set of numbers without duplicates.

Example:

```
// Init an array with set 1, 2, and 3.
int[] nums = {1,2,3};
Solution solution = new Solution(nums);

// Shuffle the array [1,2,3] and return its result. Any permutation of [1,2,3] must equally likely to be returned.
solution.shuffle();

// Resets the array back to its original configuration [1,2,3].
solution.reset();

// Returns the random shuffling of array [1,2,3].
solution.shuffle();
```


Mini Parser

Given a nested list of integers represented as a string, implement a parser to deserialize it.

Each element is either an integer, or a list -- whose elements may also be integers or other lists.

Note:
You may assume that the string is well-formed:

String is non-empty.
String does not contain white spaces.
String contains only digits 0-9, [, - ,,].

Example 1:

Given s = "324",

You should return a NestedInteger object which contains a single integer 324.

Example 2:

Given s = "[123,[456,[789]]]",

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.

Lexicographical Numbers

Given an integer n, return 1 - n in lexicographical order.

For example, given 13, return: [1,10,11,12,13,2,3,4,5,6,7,8,9].

Please optimize your algorithm to use less time and space. The input size may be as large as 5,000,000.

First Unique Character in a String

Given a string, find the first non-repeating character in it and return it's index. If it doesn't exist, return -1.

Examples:

s = "leetcode"
return 0.

s = "loveleetcode",
return 2.

Note: You may assume the string contain only lowercase letters.

Longest Absolute File Path

Suppose we abstract our file system by a string in the following manner:

The string "dir\n\tsubdir1\n\tsubdir2\n\t\tfile.ext" represents:

```
dir
    subdir1
    subdir2
        file.ext
```

The directory dir contains an empty sub-directory subdir1 and a sub-directory subdir2 containing a file file.ext.

The string "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext" represents:

```
dir
    subdir1
        file1.ext
        subsubdir1
    subdir2
        subsubdir2
            file2.ext
```

The directory dir contains two sub-directories subdir1 and subdir2. subdir1 contains a file file1.ext and an empty second-level sub-directory subsubdir1. subdir2 contains a second-level sub-directory subsubdir2 containing a file file2.ext.

We are interested in finding the longest (number of characters) absolute path to a file within our file system. For example, in the second example above, the longest absolute path is "dir/subdir2/subsubdir2/file2.ext", and its length is 32 (not including the double quotes).

Given a string representing the file system in the above format, return the length of the longest absolute path to file in the abstracted file system. If there is no file in the system, return 0.

Note:

The name of a file contains at least a . and an extension.
The name of a directory or sub-directory will not contain a ..

Time complexity required: O(n) where n is the size of the input string.

Notice that a/aa/aaa/file1.txt is not the longest file path, if there is another path aaaaaaaaaaaaaaaaaaaaa/sth.png.

Find the Difference

Given two strings s and t which consist of only lowercase letters.

String t is generated by random shuffling string s and then add one more letter at a random position.

Find the letter that was added in t.

Example:

Input:
s = "abcd"
t = "abcde"

Output:
e

Explanation:
'e' is the letter that was added.

Elimination Game

There is a list of sorted integers from 1 to n. 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 right most number and every other number from the remaining numbers.

We keep repeating the steps again, alternating left to right and right to left, until a single number remains.

Find the last number that remains starting with a list of length n.

Example:

Input:
n = 9,
1 2 3 4 5 6 7 8 9
2 4 6 8
2 6
6

Output:
6

Perfect Rectangle

Given N axis-aligned rectangles where N > 0, determine if they all together form an exact cover of a rectangular region.

Each rectangle is represented as a bottom-left point and a top-right point. For example, a unit square is represented as [1,1,2,2]. (coordinate of bottom-left point is (1, 1) and top-right point is (2, 2)).

Example 1:

```
rectangles = [
  [1,1,3,3],
  [3,1,4,2],
  [3,2,4,4],
  [1,3,2,4],
  [2,3,3,4]
]
```

Return true. All 5 rectangles together form an exact cover of a rectangular region.

Example 2:

```
rectangles = [
  [1,1,2,3],
  [1,3,2,4],
```

```
  [3,1,4,2],
  [3,2,4,4]
]
```

Return false. Because there is a gap between the two rectangular regions.

Example 3:

```
rectangles = [
  [1,1,3,3],
  [3,1,4,2],
  [1,3,2,4],
  [3,2,4,4]
]
```

Return false. Because there is a gap in the top center.

Example 4:

```
rectangles = [
  [1,1,3,3],
  [3,1,4,2],
  [1,3,2,4],
  [2,2,4,4]
]
```

Return false. Because two of the rectangles overlap with each other.

Is Subsequence

Given a string s and a string t, check if s is subsequence of t.

You may assume that there is only lower case English letters in both s and t. t is potentially a very long (length ~= 500,000) string, and s is a short string (<=100).

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, "ace" is a subsequence of "abcde" while "aec" is not).

Example 1:
s = "abc", t = "ahbgdc"

Return true.

Example 2:
s = "axc", t = "ahbgdc"

Return false.

Follow up:
If there are lots of incoming S, say S1, S2, ... , Sk where k >= 1B, and you want to check one by one to see if T has its subsequence. In this scenario, how would you change your code?

Credits:Special thanks to @pbrother for adding this problem and creating all test cases.

UTF-8 Validation

A character in UTF8 can be from 1 to 4 bytes long, subjected to the following rules:

For 1-byte character, the first bit is a 0, followed by its unicode code.
For n-bytes character, the first n-bits are all one's, the n+1 bit is 0, followed by n-1 bytes with most significant 2 bits being 10.

This is how the UTF-8 encoding would work:

```
   Char. number range  |        UTF-8 octet sequence
      (hexadecimal)    |              (binary)
   --------------------
+---------------------------------------------
   0000 0000-0000 007F | 0xxxxxxx
   0000 0080-0000 07FF | 110xxxxx 10xxxxxx
   0000 0800-0000 FFFF | 1110xxxx 10xxxxxx 10xxxxxx
   0001 0000-0010 FFFF | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
```

Given an array of integers representing the data, return whether it is a valid utf-8 encoding.

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:

data = [197, 130, 1], which represents the octet sequence: 11000101 10000010 00000001.

Return true.
It is a valid utf-8 encoding for a 2-bytes character followed by a 1-byte character.

Example 2:

data = [235, 140, 4], which represented the octet sequence: 11101011 10001100 00000100.

Return false.
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.

Decode String

Given an encoded string, return it's 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; 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 won't be input like 3a or 2[4].

Examples:

s = "3[a]2[bc]", return "aaabcbc".
s = "3[a2[c]]", return "accaccacc".
s = "2[abc]3[cd]ef", return "abcabccdcdcdef".

Longest Substring with At Least K Repeating Characters

Find the length of the longest substring T of a given string (consists of lowercase letters only) such that every character in T appears no less than k times.

Example 1:

Input:
s = "aaabb", k = 3

Output:
3

The longest substring is "aaa", as 'a' is repeated 3 times.

Example 2:

Input:
s = "ababbc", k = 2

Output:
5

The longest substring is "ababb", as 'a' is repeated 2 times and 'b' is repeated 3 times.

Rotate Function

Given an array of integers A and let n to be its length.

Assume Bk to be an array obtained by rotating the array A k positions clock-wise, we define a "rotation function" F on A as follow:

F(k) = 0 * Bk[0] + 1 * Bk[1] + ... + (n-1) * Bk[n-1].

Calculate the maximum value of F(0), F(1), ..., F(n-1).

Note:
n is guaranteed to be less than 105.

Example:

A = [4, 3, 2, 6]

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.

Integer Replacement

Given a positive integer n and you can do operations as follow:

If n is even, replace n with n/2.
If n is odd, you can replace n with either n + 1 or n - 1.

What is the minimum number of replacements needed for n to become 1?

Example 1:

Input:
8

Output:
3

Explanation:
8 -> 4 -> 2 -> 1

Example 2:

Input:
7

Output:
4

Explanation:
7 -> 8 -> 4 -> 2 -> 1
or
7 -> 6 -> 3 -> 2 -> 1

Random Pick Index

Given an array of integers 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.

Note:
The array size can be very large. Solution that uses too much extra space will not pass the judge.

Example:

```
int[] nums = new int[] {1,2,3,3,3};
Solution solution = new Solution(nums);

// pick(3) should return either index 2, 3, or 4 randomly. Each index should have equal probability of returning.
solution.pick(3);

// pick(1) should return 0. Since in the array only nums[0] is equal to 1.
solution.pick(1);
```


Evaluate Division

Equations are given in the format A / B = k, where A and B are variables represented as strings, and k is a real number (floating point number). Given some queries, return the answers. If the answer does not exist, return -1.0.

Example:
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].

The input is: vector<pair<string, string>> equations, vector<double>& values, vector<pair<string, string>> queries , where equations.size() == values.size(), and the values are positive. This represents the equations. Return vector<double>.

According to the example above:
equations = [["a", "b"], ["b", "c"]],
values = [2.0, 3.0],
queries = [["a", "c"], ["b", "a"], ["a", "e"], ["a", "a"], ["x", "x"]].

The input is always valid. You may assume that evaluating the queries will result in no division by zero and there is no contradiction.

Nth Digit

Find the nth digit of the infinite integer sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

Note:
n is positive and will fit within the range of a 32-bit signed integer (n < 231).

Example 1:

Input:
3

Output:
3

Example 2:

Input:
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.

Binary Watch

A binary watch has 4 LEDs on the top which represent the hours (0-11), and the 6 LEDs on the bottom represent the minutes (0-59). Each LED represents a zero or one, with the least significant bit on the right.

For example, the above binary watch reads "3:25".

Given a non-negative integer n which represents the number of LEDs that are currently on, return all possible times the watch could represent.

Example:
Input: n = 1Return: ["1:00", "2:00", "4:00", "8:00", "0:01", "0:02", "0:04", "0:08", "0:16", "0:32"]

Note:

The order of output does not matter.
The hour must not contain a leading zero, for example "01:00" is not valid, it should be "1:00".
The minute must be consist of two digits and may contain a leading zero, for example "10:2" is not valid, it should be "10:02".

Remove K Digits

Given a non-negative integer num represented as a string, remove k digits from the number so that the new number is the smallest possible.

Note:

The length of num is less than 10002 and will be ≥ k.
The given num does not contain any leading zero.

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.

Frog Jump

A frog is crossing a river. The river is divided into x 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 is able to cross the river by landing on the last stone. Initially, the frog is on the first stone and assume the first jump must be 1 unit.

If the frog's last jump was k units, then its next jump must be either k - 1, k, or k + 1 units. Note that the frog can only jump in the forward direction.

Note:

The number of stones is ≥ 2 and is < 1,100.
Each stone's position will be a non-negative integer < 231.

The first stone's position is always 0.

Example 1:

[0,1,3,5,6,8,12,17]

There are a total of 8 stones.
The first stone at the 0th unit, second stone at the 1st unit,
third stone at the 3rd unit, and so on...
The last stone at the 17th unit.

Return true. 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:

[0,1,2,3,4,8,9,11]

Return false. There is no way to jump to the last stone as
the gap between the 5th and 6th stone is too large.

Sum of Left Leaves

Find the sum of all left leaves in a given binary tree.

Example:

```
    3
   / \
  9  20
    /  \
   15   7
```

There are two left leaves in the binary tree, with values 9 and 15
respectively. Return 24.

Convert a Number to Hexadecimal

Given an integer, write an algorithm to convert it to hexadecimal.
For negative integer, two’s complement method is used.

Note:

All letters in hexadecimal (a-f) must be in lowercase.
The hexadecimal string must not contain extra leading 0s. If the number is zero, it is represented by a single zero character '0'; otherwise, the first character in the hexadecimal string will not be the zero character.
The given number is guaranteed to fit within the range of a 32-bit signed integer.
You must not use any method provided by the library which converts/formats the number to hex directly.

Example 1:

Input:
26

Output:
"1a"

Example 2:

Input:
-1

Output:
"ffffffff"

Queue Reconstruction by Height

Suppose you have a random list of people standing in a queue. Each person is described by a pair of integers (h, k), where h is the height of the person and k is the number of people in front of this person who have a height greater than or equal to h. Write an algorithm to reconstruct the queue.

Note:
The number of people is less than 1,100.

Example

Input:
[[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]]

Trapping Rain Water II

Given an m x n matrix of positive integers representing the height of each unit cell in a 2D elevation map, compute the volume of water it is able to trap after raining.

Note:
Both m and n are less than 110. The height of each unit cell is greater than 0 and is less than 20,000.

Example:

Given the following 3x6 height map:
[
  [1,4,3,1,3,2],
  [3,2,1,3,2,4],
  [2,3,3,2,3,1]
]

Return 4.

The above image represents the elevation map [[1,4,3,1,3,2], [3,2,1,3,2,4],[2,3,3,2,3,1]] before the rain.

After the rain, water are trapped between the blocks. The total volume of water trapped is 4.

Longest Palindrome

Given a string which consists of lowercase or uppercase letters, find the length of the longest palindromes that can be built with those letters.

This is case sensitive, for example "Aa" is not considered a palindrome here.

Note:
Assume the length of given string will not exceed 1,010.

Example:

Input:
"abccccdd"

Output:
7

Explanation:
One longest palindrome that can be built is "dccaccd", whose length is 7.

Split Array Largest Sum

Given an array which consists of non-negative integers and an integer m, you can split the array into m non-empty continuous subarrays. Write an algorithm to minimize the largest sum among these m subarrays.

Note:
If n is the length of array, assume the following constraints are satisfied:

1 ≤ n ≤ 1000
1 ≤ m ≤ min(50, n)

Examples:

Input:
nums = [7,2,5,10,8]
m = 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.

Fizz Buzz

Write a program that outputs the string representation of numbers from 1 to n.

But for multiples of three it should output “Fizz” instead of the number and for the multiples of five output “Buzz”. For numbers which are multiples of both three and five output “FizzBuzz”.

Example:

n = 15,

Return:

```
[
    "1",
    "2",
    "Fizz",
    "4",
    "Buzz",
    "Fizz",
    "7",
    "8",
    "Fizz",
    "Buzz",
    "11",
    "Fizz",
    "13",
    "14",
    "FizzBuzz"
]
```

Arithmetic Slices

A sequence of number 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, these are arithmetic sequence:
1, 3, 5, 7, 9
7, 7, 7, 7
3, -1, -5, -9

The following sequence is not arithmetic. 1, 1, 2, 5, 7

A zero-indexed array A consisting of N numbers is given. A slice of that array is any pair of integers (P, Q) such that 0 <= P < Q < N.

A slice (P, Q) of array A is called arithmetic if the sequence:
    A[P], A[p + 1], ..., A[Q - 1], A[Q] is arithmetic. In

particular, this means that P + 1 < Q.

The function should return the number of arithmetic slices in the array A.

Example:

A = [1, 2, 3, 4]

return: 3, for 3 arithmetic slices in A: [1, 2, 3], [2, 3, 4] and [1, 2, 3, 4] itself.

Third Maximum Number

Given a non-empty array of integers, return the third maximum number in this array. If it does not exist, return the maximum number. The time complexity must be in O(n).

Example 1:

Input: [3, 2, 1]

Output: 1

Explanation: The third maximum is 1.

Example 2:

Input: [1, 2]

Output: 2

Explanation: The third maximum does not exist, so the maximum (2) is returned instead.

Example 3:

Input: [2, 2, 3, 1]

Output: 1

Explanation: Note that the third maximum here means the third maximum distinct number.
Both numbers with value 2 are both considered as second maximum.

Add Strings

Given two non-negative integers num1 and num2 represented as string, return the sum of num1 and num2.

Note:

The length of both num1 and num2 is < 5100.
Both num1 and num2 contains only digits 0-9.
Both num1 and num2 does not contain any leading zero.
You must not use any built-in BigInteger library or convert the inputs to integer directly.

Partition Equal Subset Sum

Given a non-empty array containing only positive integers, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

Note:

Each of the array element will not exceed 100.
The array size will not exceed 200.

Example 1:

Input: [1, 5, 11, 5]

Output: true

Explanation: The array can be partitioned as [1, 5, 5] and [11].

Example 2:

Input: [1, 2, 3, 5]

Output: false

Explanation: The array cannot be partitioned into equal sum subsets.

Pacific Atlantic Water Flow

Given an m x n matrix of non-negative integers representing the height of each unit cell in a continent, the "Pacific ocean" touches the left and top edges of the matrix and the "Atlantic ocean" touches the right and bottom edges.

Water can only flow in four directions (up, down, left, or right) from a cell to another one with height equal or lower.

Find the list of grid coordinates where water can flow to both the Pacific and Atlantic ocean.

Note:

The order of returned grid coordinates does not matter.
Both m and n are less than 150.

Example:

Given the following 5x5 matrix:

```
  Pacific ~   ~   ~   ~   ~
       ~  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  *
          *   *   *   *   * Atlantic
```

Return:

[[0, 4], [1, 3], [1, 4], [2, 2], [3, 0], [3, 1], [4, 0]] (positions with parentheses in above matrix).

Battleships in a Board

Given an 2D board, count how many battleships are in it. The battleships are represented with 'X's, empty slots are represented with '.'s. You may assume the following rules:

You receive a valid board, made of only battleships or empty slots.
Battleships can only be placed horizontally or vertically. In other words, they can only be made of the shape 1xN (1 row, N columns) or Nx1 (N rows, 1 column), where N can be of any size.
At least one horizontal or vertical cell separates between two battleships - there are no adjacent battleships.

Example:

X..X
...X
...X

In the above board there are 2 battleships.

Invalid Example:
...X
XXXX
...X

This is an invalid board that you will not receive - as battleships will always have a cell separating between them.

Follow up:Could you do it in one-pass, using only O(1) extra memory and without modifying the value of the board?

Strong Password Checker

A password is considered strong if below conditions are all met:

 It has at least 6 characters and at most 20 characters.
 It must contain at least one lowercase letter, at least one uppercase letter, and at least one digit.
 It must NOT contain three repeating characters in a row ("...aaa..." is weak, but "...aa...a..." is strong, assuming other conditions are met).

Write a function strongPasswordChecker(s), that takes a string s as input, and return the MINIMUM change required to make s a strong password. If s is already strong, return 0.

Insertion, deletion or replace of any one character are all considered as one change.

Maximum XOR of Two Numbers in an Array

Given a non-empty array of numbers, a0, a1, a2, … , an-1, where 0 ≤ ai < 231.

Find the maximum result of ai XOR aj, where 0 ≤ i, j < n.

Could you do this in O(n) runtime?

Example:

Input: [3, 10, 5, 25, 2, 8]

Output: 28

Explanation: The maximum result is 5 ^ 25 = 28.

Reconstruct Original Digits from English

Given a non-empty string containing an out-of-order English representation of digits 0-9, output the digits in ascending order.

Note:

Input contains only lowercase English letters.
Input is guaranteed to be valid and can be transformed to its original digits. That means invalid inputs such as "abc" or "zerone" are not permitted.
Input length is less than 50,000.

Example 1:

Input: "owoztneoer"

Output: "012"

Example 2:

Input: "fviefuro"

Output: "45"

Longest Repeating Character Replacement

Given a string that consists of only uppercase English letters, you can replace any letter in the string with another letter at most k times. Find the length of a longest substring containing all repeating letters you can get after performing the above operations.

Note:
Both the string's length and k will not exceed 104.

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.

All O`one Data Structure

Implement a data structure supporting the following operations:

Inc(Key) - Inserts a new key with value 1. Or increments an existing key by 1. Key is guaranteed to be a non-empty string.
Dec(Key) - If Key's value is 1, remove it from the data structure. Otherwise decrements an existing key by 1. If the key does not exist, this function does nothing. Key is guaranteed to be a non-empty string.
GetMaxKey() - Returns one of the keys with maximal value. If no element exists, return an empty string "".
GetMinKey() - Returns one of the keys with minimal value. If no element exists, return an empty string "".

Challenge: Perform all these in O(1) time complexity.

Number of Segments in a String

Count the number of segments in a string, where a segment is defined to be a contiguous sequence of non-space characters.

Please note that the string does not contain any non-printable characters.

Example:

Input: "Hello, my name is John"
Output: 5

Non-overlapping Intervals

Given a collection of intervals, find the minimum number of intervals you need to remove to make the rest of the intervals non-overlapping.

Note:

You may assume the interval's end point is always bigger than its start point.
Intervals like [1,2] and [2,3] have borders "touching" but they don't overlap each other.

Example 1:

Input: [[1,2], [2,3], [3,4], [1,3]]

Output: 1

Explanation: [1,3] can be removed and the rest of intervals are non-overlapping.

Example 2:

Input: [[1,2], [1,2], [1,2]]

Output: 2

Explanation: You need to remove two [1,2] to make the rest of intervals non-overlapping.

Example 3:

Input: [[1,2], [2,3]]

Output: 0

Explanation: You don't need to remove any of the intervals since they're already non-overlapping.

Find Right Interval

Given a set of intervals, for each of the interval i, check if there exists an interval j whose start point is bigger than or equal to the end point of the interval i, which can be called that j is on the "right" of i.

For any interval i, you need to store the minimum interval j's index, which means that the interval j has the minimum start point to build the "right" relationship for interval i. If the interval j doesn't exist, store -1 for the interval i. Finally, you need output the stored value of each interval as an array.

Note:

You may assume the interval's end point is always bigger than its start point.
You may assume none of these intervals have the same start point.

Example 1:

Input: [[1,2]]

Output: [-1]

Explanation: There is only one interval in the collection, so it outputs -1.

Example 2:

Input: [[3,4], [2,3], [1,2]]

Output: [-1, 0, 1]

Explanation: There is no satisfied "right" interval for [3,4].
For [2,3], the interval [3,4] has minimum-"right" start point;
For [1,2], the interval [2,3] has minimum-"right" start point.

Example 3:

Input: [[1,4], [2,3], [3,4]]

Output: [-1, 2, -1]

Explanation: There is no satisfied "right" interval for [1,4] and [3,4].
For [2,3], the interval [3,4] has minimum-"right" start point.

Path Sum III

You are given a binary tree in which each node contains an integer value.

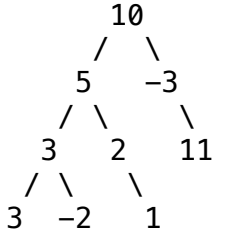
Find the number of paths that sum to a given value.

The path does not need to start or end at the root or a leaf, but it must go downwards
(traveling only from parent nodes to child nodes).

The tree has no more than 1,000 nodes and the values are in the range -1,000,000 to 1,000,000.

Example:

root = [10,5,-3,3,2,null,11,3,-2,null,1], sum = 8



Return 3. The paths that sum to 8 are:

1. 5 -> 3
2. 5 -> 2 -> 1
3. -3 -> 11

Find All Anagrams in a String

Given a string s and a non-empty string p, find all the start

indices of p's anagrams in s.

Strings consists of lowercase English letters only and the length of both strings s and p will not be larger than 20,100.

The order of output does not matter.

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".

K-th Smallest in Lexicographical Order

Given integers n and k, find the lexicographically k-th smallest integer in the range from 1 to n.

Note: 1 ≤ k ≤ n ≤ 109.

Example:

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.

Arranging Coins

You have a total of n coins that you want to form in a staircase shape, where every k-th row must have exactly k coins.

Given n, find the total number of full staircase rows that can be formed.

n is a non-negative integer and fits within the range of a 32-bit signed integer.

Example 1:

n = 5

The coins can form the following rows:
¤
¤ ¤
¤ ¤

Because the 3rd row is incomplete, we return 2.

Example 2:

n = 8

The coins can form the following rows:
¤
¤ ¤
¤ ¤ ¤
¤ ¤

Because the 4th row is incomplete, we return 3.

Find All Duplicates in an Array

Given an array of integers, 1 ≤ a[i] ≤ n (n = size of array), some elements appear twice and others appear once.

Find all the elements that appear twice in this array.

Could you do it without extra space and in O(n) runtime?

Example:

Input:
[4,3,2,7,8,2,3,1]

Output:
[2,3]

Add Two Numbers II

You are given two non-empty linked lists representing two non-negative integers. The most significant digit comes first and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Follow up:
What if you cannot modify the input lists? In other words, reversing the lists is not allowed.

Example:

Input: (7 -> 2 -> 4 -> 3) + (5 -> 6 -> 4)
Output: 7 -> 8 -> 0 -> 7

Arithmetic Slices II - Subsequence

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, 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

A zero-indexed array A consisting of N numbers is given. A subsequence slice of that array is any sequence of integers (P0, P1, ..., Pk) such that 0 ≤ P0 < P1 < ... < Pk < N.

A subsequence slice (P0, P1, ..., Pk) of array A is called arithmetic if the sequence A[P0], A[P1], ..., A[Pk-1], A[Pk] is arithmetic. In particular, this means that k ≥ 2.

The function should return the number of arithmetic subsequence slices in the array A.

The input contains N integers. Every integer is in the range of -231 and 231-1 and 0 ≤ N ≤ 1000. The output is guaranteed to be less than 231-1.

Example:

Input: [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]

Number of Boomerangs

Given n points in the plane that are all pairwise distinct, 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).

Find the number of boomerangs. You may assume that n will be at most 500 and coordinates of points are all in the range [-10000, 10000] (inclusive).

Example:

Input:
[[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]]

Find All Numbers Disappeared in an Array

Given an array of integers where 1 ≤ a[i] ≤ n (n = size of array), some elements appear twice and others appear once.

Find all the elements of [1, n] inclusive that do not appear in this array.

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

Example:

Input:
[4,3,2,7,8,2,3,1]

Output:
[5,6]

Serialize and Deserialize BST

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 search tree. There is no restriction on how your serialization/ deserialization algorithm should work. You just 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.

Note: Do not use class member/global/static variables to store states. Your serialize and deserialize algorithms should be stateless.

Delete Node in a BST

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.

Note: Time complexity should be O(height of tree).

Example:

```
root = [5,3,6,2,4,null,7]
key = 3

    5
   / \
  3   6
 / \   \
2   4   7
```

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 following BST.

```
    5
   / \
  4   6
 /     \
2       7
```

Another valid answer is [5,2,6,null,4,null,7].

```
    5
   / \
  2   6
   \   \
    4   7
```


Sort Characters By Frequency

Given a string, sort it in decreasing order based on the frequency of characters.

Example 1:

Input:
"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:
"cccaaa"

Output:
"cccaaa"

Explanation:
Both 'c' and 'a' appear three times, so "aaaccc" is also a valid answer.
Note that "cacaca" is incorrect, as the same characters must be together.

Example 3:

Input:
"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.

Minimum Moves to Equal Array Elements

Given a non-empty integer array of size n, find the minimum number of moves required to make all array elements equal, where a move is incrementing n - 1 elements by 1.

Example:

Input:
[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]

4Sum II

Given four lists A, B, C, D of integer values, compute how many tuples (i, j, k, l) there are such that A[i] + B[j] + C[k] + D[l] is zero.

To make problem a bit easier, all A, B, C, D have same length of N where 0 ≤ N ≤ 500. All integers are in the range of -228 to 228 - 1 and the result is guaranteed to be at most 231 - 1.

Example:

Input:
A = [1, 2]
B = [-2,-1]
C = [-1, 2]
D = [0, 2]

Output:
2

Explanation:
The two tuples are:
1. (0, 0, 0, 1) -> A[0] + B[0] + C[0] + D[1] = 1 + (-2) + (-1) + 2 = 0
2. (1, 1, 0, 0) -> A[1] + B[1] + C[0] + D[0] = 2 + (-1) + (-1) + 0 = 0

Assign Cookies

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 gi, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size sj. If sj >= gi, 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.

Note:
You may assume the greed factor is always positive.
You cannot assign more than one cookie to one child.

Example 1:

Input: [1,2,3], [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: [1,2], [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.

---
132 Pattern

Given a sequence of n integers a1, a2, ..., an, a 132 pattern is a subsequence ai, aj, ak such
that i < j < k and ai < ak < aj. Design an algorithm that takes a list of n numbers as input and checks whether there is a 132 pattern in the list.

Note: n will be less than 15,000.

Example 1:

Input: [1, 2, 3, 4]

Output: False

Explanation: There is no 132 pattern in the sequence.

Example 2:

Input: [3, 1, 4, 2]

Output: True

Explanation: There is a 132 pattern in the sequence: [1, 4, 2].

Example 3:

Input: [-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].

Repeated Substring Pattern

Given a non-empty string check if it can be constructed by taking a substring of it and appending multiple copies of the substring together. You may assume the given string consists of lowercase English letters only and its length will not exceed 10000.

Example 1:

Input: "abab"

Output: True

Explanation: It's the substring "ab" twice.

Example 2:

Input: "aba"

Output: False

Example 3:

Input: "abcabcabcabc"

Output: True

Explanation: It's the substring "abc" four times. (And the substring "abcabc" twice.)

LFU Cache

Design and implement a data structure for Least Frequently Used (LFU) cache. It should support the following operations: get and put.

get(key) - Get the value (will always be positive) of the key if the key exists in the cache, otherwise return -1.
put(key, value) - Set or insert the value if the key is not already present. When the cache reaches its capacity, it should invalidate the least frequently used item before inserting a new item. For the purpose of this problem, when there is a tie (i.e., two or more keys that have the same frequency), the least recently used key would be evicted.

Follow up:
Could you do both operations in O(1) time complexity?

Example:

```
LFUCache cache = new LFUCache( 2 /* capacity */ );

cache.put(1, 1);
cache.put(2, 2);
cache.get(1);       // returns 1
cache.put(3, 3);    // evicts key 2
cache.get(2);       // returns -1 (not found)
cache.get(3);       // returns 3.
cache.put(4, 4);    // evicts key 1.
cache.get(1);       // returns -1 (not found)
cache.get(3);       // returns 3
cache.get(4);       // returns 4
```


Hamming Distance

The Hamming distance between two integers is the number of positions at which the corresponding bits are different.

Given two integers x and y, calculate the Hamming distance.

Note:
0 ≤ x, y < 231.

Example:

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.

Minimum Moves to Equal Array Elements II

Given a non-empty integer array, find the minimum number of moves required to make all array elements equal, where a move is incrementing a selected element by 1 or decrementing a selected element by 1.

You may assume the array's length is at most 10,000.

Example:

Input:
[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]

Island Perimeter

You are given a map in form of a two-dimensional integer grid where 1 represents land and 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" (water

inside that 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:

```
[[0,1,0,0],
 [1,1,1,0],
 [0,1,0,0],
 [1,1,0,0]]
```

Answer: 16
Explanation: The perimeter is the 16 yellow stripes in the image below:

Can I Win

In the "100 game," two players take turns adding, to a running total, any integer from 1..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 of 1..15 without replacement until they reach a total >= 100.

Given an integer maxChoosableInteger and another integer desiredTotal, determine if the first player to move can force a win, assuming both players play optimally.

You can always assume that maxChoosableInteger will not be larger than 20 and desiredTotal will not be larger than 300.

Example

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.

Count The Repetitions

Define S = [s,n] as the string S which consists of n connected strings s. For example, ["abc", 3] ="abcabcabc".
On the other hand, 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, “abc” can be obtained from “abdbec” based on our definition, but it can not be obtained from “acbbe”.
You are given two non-empty strings s1 and s2 (each at most 100 characters long) and two integers 0 ≤ n1 ≤ 106 and 1 ≤ n2 ≤ 106. Now consider the strings S1 and S2, where S1=[s1,n1] and S2=[s2,n2]. Find the maximum integer M such that [S2,M] can be obtained from S1.

Example:

Input:
s1="acb", n1=4
s2="ab", n2=2

Return:
2

Unique Substrings in Wraparound String

Consider the string s to be the infinite wraparound string of "abcdefghijklmnopqrstuvwxyz", so s will look like this: "...zabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcd....".

Now we have another string p. Your job is to find out how many unique non-empty substrings of p are present in s. In particular, your input is the string p and you need to output the number of different non-empty substrings of p in the string s.

Note: p consists of only lowercase English letters and the size of p might be over 10000.

Example 1:

Input: "a"
Output: 1

Explanation: Only the substring "a" of string "a" is in the string s.

Example 2:

Input: "cac"
Output: 2
Explanation: There are two substrings "a", "c" of string "cac" in the string s.

Example 3:

Input: "zab"
Output: 6
Explanation: There are six substrings "z", "a", "b", "za", "ab", "zab" of string "zab" in the string s.

Validate IP Address

Write a function to check whether an input string is a valid IPv4 address or IPv6 address or neither.

IPv4 addresses are canonically represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, separated by dots ("."), e.g.,172.16.254.1;

Besides, leading zeros in the IPv4 is invalid. For example, the address 172.16.254.01 is invalid.

IPv6 addresses are represented as eight groups of four hexadecimal digits, each group representing 16 bits. The groups are separated by colons (":"). For example, the address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 is a valid one. Also, we could omit some leading zeros among four hexadecimal digits and some low-case characters in the address to upper-case ones, so 2001:db8:85a3:0:0:8A2E:0370:7334 is also a valid IPv6 address(Omit leading zeros and using upper cases).

However, we don't replace a consecutive group of zero value with a single empty group using two consecutive colons (::) to pursue simplicity. For example, 2001:0db8:85a3::8A2E:0370:7334 is an invalid IPv6 address.

Besides, extra leading zeros in the IPv6 is also invalid. For example, the address 02001:0db8:85a3:0000:0000:8a2e:0370:7334 is invalid.

Note:
You may assume there is no extra space or special characters in the input string.

Example 1:

Input: "172.16.254.1"

Output: "IPv4"

Explanation: This is a valid IPv4 address, return "IPv4".

Example 2:

Input: "2001:0db8:85a3:0:0:8A2E:0370:7334"

Output: "IPv6"

Explanation: This is a valid IPv6 address, return "IPv6".

Example 3:

Input: "256.256.256.256"

Output: "Neither"

Explanation: This is neither a IPv4 address nor a IPv6 address.

Concatenated Words

Given a list of words (without duplicates), please write a program that returns all 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 in the given array.

Example:

Input:
["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".

Note:

The number of elements of the given array will not exceed 10,000
The length sum of elements in the given array will not exceed 600,000.
All the input string will only include lower case letters.
The returned elements order does not matter.

Matchsticks to Square

Remember the story of Little Match Girl? By now, you know exactly what matchsticks the little match girl has, please find out a way you can make one square by using up all those matchsticks. You should not break any stick, but you can link them up, and each matchstick must be used exactly one time.

 Your input will be several matchsticks the girl has, represented with their stick length. Your output will either be true or false, to represent whether you could make one square using all the matchsticks the little match girl has.

Example 1:

Input: [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: [3,3,3,3,4]
Output: false

Explanation: You cannot find a way to form a square with all the matchsticks.

Note:

The length sum of the given matchsticks is in the range of 0 to 10^9.
The length of the given matchstick array will not exceed 15.

Ones and Zeroes

In the computer world, use restricted resource you have to generate maximum benefit is what we always want to pursue.
For now, suppose you are a dominator of m 0s and n 1s respectively. On the other hand, there is an array with strings consisting of only 0s and 1s.

Now your task is to find the maximum number of strings that you can form with given m 0s and n 1s. Each 0 and 1 can be used at most once.

Note:

The given numbers of 0s and 1s will both not exceed 100
The size of given string array won't exceed 600.

Example 1:

Input: Array = {"10", "0001", "111001", "1", "0"}, m = 5, n = 3
Output: 4

Explanation: This are totally 4 strings can be formed by the using of 5 0s and 3 1s, which are “10,”0001”,”1”,”0”

Example 2:

Input: Array = {"10", "0", "1"}, m = 1, n = 1
Output: 2

Explanation: You could form "10", but then you'd have nothing left. Better form "0" and "1".

Heaters

Winter is coming! Your first job during the contest is to design a standard heater with fixed warm radius to warm all the houses.

Now, you are given positions of houses and heaters on a horizontal line, find out minimum radius of heaters so that all houses could be covered by those heaters.

So, your input will be the positions of houses and heaters seperately, and your expected output will be the minimum radius standard of heaters.

Note:

Numbers of houses and heaters you are given are non-negative and will not exceed 25000.
Positions of houses and heaters you are given are non-negative and will not exceed 10^9.
As long as a house is in the heaters' warm radius range, it can be warmed.
All the heaters follow your radius standard and the warm radius will the same.

Example 1:

Input: [1,2,3],[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: [1,2,3,4],[1,4]
Output: 1
Explanation: The two heater was placed in the position 1 and 4. We need to use radius 1 standard, then all the houses can be warmed.

Number Complement

Given a positive integer, output its complement number. The complement strategy is to flip the bits of its binary representation.

Note:

The given integer is guaranteed to fit within the range of a 32-bit signed integer.
You could assume no leading zero bit in the integer’s binary representation.

Example 1:

Input: 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: 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.

Total Hamming Distance

The Hamming distance between two integers is the number of positions at which the corresponding bits are different.

Now your job is to find the total Hamming distance between all pairs of the given numbers.

Example:

Input: 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). So the answer will be: HammingDistance(4, 14) + HammingDistance(4, 2) + HammingDistance(14, 2) = 2 + 2 + 2 = 6.

Note:

Elements of the given array are in the range of 0 to 10^9
Length of the array will not exceed 10^4.

Sliding Window Median

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 value.
Examples:
[2,3,4] , the median is 3
[2,3], the median is (2 + 3) / 2 = 2.5

Given an array 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. Your job is to output the median array for each window in the original array.

For example,
Given nums = [1,3,-1,-3,5,3,6,7], and k = 3.

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

Therefore, return the median sliding window as [1,-1,-1,3,5,6].

Note:
You may assume k is always valid, ie: 1 ≤ k ≤ input array's size for non-empty array.

Magical String

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 string S is the following:
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 above is the S itself.

Given an integer N as input, return the number of '1's in the first N number in the magical string S.

Note:
N will not exceed 100,000.

Example 1:

Input: 6
Output: 3
Explanation: The first 6 elements of magical string S is "12211" and it contains three 1's, so return 3.

License Key Formatting

Now you are given a string S, which represents a software license key which we would like to format. The string S is composed of alphanumerical characters and dashes. The dashes split the alphanumerical characters within the string into groups. (i.e. if there are M dashes, the string is split into M+1 groups). The dashes

in the given string are possibly misplaced.

We want each group of characters to be of length K (except for possibly the first group, which could be shorter, but still must contain at least one character). To satisfy this requirement, we will reinsert dashes. Additionally, all the lower case letters in the string must be converted to upper case.

So, you are given a non-empty string S, representing a license key to format, and an integer K. And you need to return the license key formatted according to the description above.

Example 1:

Input: S = "2-4A0r7-4k", K = 4

Output: "24A0-R74K"

Explanation: The string S has been split into two parts, each part has 4 characters.

Example 2:

Input: S = "2-4A0r7-4k", K = 3

Output: "24-A0R-74K"

Explanation: The string S has been split into three parts, each part has 3 characters except the first part as it could be shorter as said above.

Note:

The length of string S will not exceed 12,000, and K is a positive integer.
String S consists only of alphanumerical characters (a-z and/or A-Z and/or 0-9) and dashes(-).
String S is non-empty.

Smallest Good Base

For an integer n, we call k>=2 a good base of n, if all digits of n base k are 1.
Now given a string representing n, you should return the smallest good base of n in string format.

Example 1:

Input: "13"
Output: "3"
Explanation: 13 base 3 is 111.

Example 2:

Input: "4681"
Output: "8"
Explanation: 4681 base 8 is 11111.

Example 3:

Input: "1000000000000000000"
Output: "999999999999999999"
Explanation: 1000000000000000000 base 999999999999999999 is 11.

Note:

The range of n is [3, 10^18].
The string representing n is always valid and will not have leading zeros.

Max Consecutive Ones

Given a binary array, find the maximum number of consecutive 1s in this array.

Example 1:

Input: [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.

Note:

The input array will only contain 0 and 1.
The length of input array is a positive integer and will not exceed

10,000

Predict the Winner

Given an array of scores that are non-negative integers. Player 1 picks one of the numbers from either end of the array followed by the player 2 and then player 1 and so on. Each time a player picks a number, that number will not be available for the next player. This continues until all the scores have been chosen. The player with the maximum score wins.

Given an array of scores, predict whether player 1 is the winner. You can assume each player plays to maximize his score.

Example 1:

Input: [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: [1, 5, 233, 7]
Output: True
Explanation: Player 1 first chooses 1. Then player 2 have 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.

Note:

1 <= length of the array <= 20.
Any scores in the given array are non-negative integers and will not exceed 10,000,000.
If the scores of both players are equal, then player 1 is still the winner.

Zuma Game

Think about Zuma Game. You have a row of balls on the table, colored red(R), yellow(Y), blue(B), green(G), and white(W). You also have several balls in your hand.

Each time, you may choose a ball in your hand, and insert it into the row (including the leftmost place and rightmost place). Then, if there is a group of 3 or more balls in the same color touching, remove these balls. Keep doing this until no more balls can be removed.

Find the minimal balls you have to insert to remove all the balls on the table. If you cannot remove all the balls, output -1.

Examples:
Input: "WRRBBW", "RB"
Output: -1
Explanation: WRRBBW -> WRR[R]BBW -> WBBW -> WBB[B]W -> WW

Input: "WWRRBBWW", "WRBRW"
Output: 2
Explanation: WWRRBBWW -> WWRR[R]BBWW -> WWBBWW -> WWBB[B]WW -> WWWW -> empty

Input:"G", "GGGGG"
Output: 2
Explanation: G -> G[G] -> GG[G] -> empty

Input: "RBYYBBRRB", "YRBGB"
Output: 3
Explanation: RBYYBBRRB -> RBYY[Y]BBRRB -> RBBBRRB -> RRRB -> B -> B[B] -> BB[B] -> empty

Note:

You may assume that the initial row of balls on the table won’t have any 3 or more consecutive balls with the same color.
The number of balls on the table won't exceed 20, and the string represents these balls is called "board" in the input.
The number of balls in your hand won't exceed 5, and the string represents these balls is called "hand" in the input.
Both input strings will be non-empty and only contain characters 'R','Y','B','G','W'.

Increasing Subsequences

Given an integer array, your task is to find all the different

possible increasing subsequences of the given array, and the length of an increasing subsequence should be at least 2 .

Example:

Input: [4, 6, 7, 7]
Output: [[4, 6], [4, 7], [4, 6, 7], [4, 6, 7, 7], [6, 7], [6, 7, 7], [7,7], [4,7,7]]

Note:

The length of the given array will not exceed 15.
The range of integer in the given array is [-100,100].
The given array may contain duplicates, and two equal integers should also be considered as a special case of increasing sequence.

Construct the Rectangle

For a web developer, it is very important 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:
1. The area of the rectangular web page you designed must equal to the given target area.
2. The width W should not be larger than the length L, which means L >= W.
3. The difference between length L and width W should be as small as possible.

You need to output the length L and the width W of the web page you designed in sequence.

Example:

Input: 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.

Note:

The given area won't exceed 10,000,000 and is a positive integer
The web page's width and length you designed must be positive integers.

Reverse Pairs

Given an array nums, we call (i, j) an important reverse pair if i < j and nums[i] > 2*nums[j].

You need to return the number of important reverse pairs in the given array.

Example1:

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

Example2:

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

Note:

The length of the given array will not exceed 50,000.
All the numbers in the input array are in the range of 32-bit integer.

Target Sum

You are given a list of non-negative integers, a1, a2, ..., an, and a target, S. Now you have 2 symbols + and -. For each integer, you should choose one from + and - as its new symbol.

Find out how many ways to assign symbols to make sum of integers equal to target S.

Example 1:

Input: nums is [1, 1, 1, 1, 1], S is 3.
Output: 5
Explanation:

-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

There are 5 ways to assign symbols to make the sum of nums be target 3.

Note:

The length of the given array is positive and will not exceed 20.
The sum of elements in the given array will not exceed 1000.
Your output answer is guaranteed to be fitted in a 32-bit integer.

Teemo Attacking

In LLP world, there is a hero called Teemo and his attacking can make his enemy Ashe be in poisoned condition. Now, given the Teemo's attacking ascending time series towards Ashe and the poisoning time duration per Teemo's attacking, you need to output the total time that Ashe is in poisoned condition.

You may assume that Teemo attacks at the very beginning of a specific time point, and makes Ashe be in poisoned condition immediately.

Example 1:

Input: [1,4], 2
Output: 4
Explanation: At time point 1, Teemo starts attacking Ashe and makes Ashe be poisoned immediately. This poisoned status will last 2 seconds until the end of time point 2. And at time point 4, Teemo attacks Ashe again, and causes Ashe to be in poisoned status for another 2 seconds. So you finally need to output 4.

Example 2:

Input: [1,2], 2
Output: 3
Explanation: At time point 1, Teemo starts attacking Ashe and makes Ashe be poisoned. This poisoned status will last 2 seconds until the

end of time point 2. However, at the beginning of time point 2, Teemo attacks Ashe again who is already in poisoned status. Since the poisoned status won't add up together, though the second poisoning attack will still work at time point 2, it will stop at the end of time point 3. So you finally need to output 3.

Note:

You may assume the length of given time series array won't exceed 10000.
You may assume the numbers in the Teemo's attacking time series and his poisoning time duration per attacking are non-negative integers, which won't exceed 10,000,000.

Next Greater Element I

You are given two arrays (without duplicates) nums1 and nums2 where nums1’s elements are subset of nums2. Find all the next greater numbers for nums1's elements in the corresponding places of nums2.

The Next Greater Number of a number x in nums1 is the first greater number to its right in nums2. If it does not exist, output -1 for this number.

Example 1:

Input: nums1 = [4,1,2], nums2 = [1,3,4,2].
Output: [-1,3,-1]
Explanation:
    For number 4 in the first array, you cannot find the next greater number for it in the second array, so output -1.
    For number 1 in the first array, the next greater number for it in the second array is 3.
    For number 2 in the first array, there is no next greater number for it in the second array, so output -1.

Example 2:

Input: nums1 = [2,4], nums2 = [1,2,3,4].
Output: [3,-1]
Explanation:
    For number 2 in the first array, the next greater number for it

in the second array is 3.
    For number 4 in the first array, there is no next greater number for it in the second array, so output -1.

Note:

All elements in nums1 and nums2 are unique.
The length of both nums1 and nums2 would not exceed 1000.

Diagonal Traverse

Given a matrix of M x N elements (M rows, N columns), return all elements of the matrix in diagonal order as shown in the below image.

Example:

Input:
[
 [1, 2, 3],
 [4, 5, 6],
 [7, 8, 9]
]
Output: [1,2,4,7,5,3,6,8,9]
Explanation:

Note:

The total number of elements of the given matrix will not exceed 10,000.

Keyboard Row

Given a List of words, return the words that can be typed using letters of alphabet on only one row's of American keyboard like the image below.

Example 1:

Input: ["Hello", "Alaska", "Dad", "Peace"]
Output: ["Alaska", "Dad"]

Note:

You may use one character in the keyboard more than once.
You may assume the input string will only contain letters of alphabet.

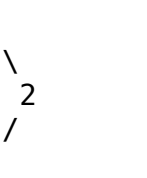
Find Mode in Binary Search Tree

Given a binary search tree (BST) with duplicates, find all the mode(s) (the most frequently occurred element) in the given BST.

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.

For example:
Given BST [1,null,2,2],



return [2].

Note:
If a tree has more than one mode, you can return them in any order.

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

IPO

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 several projects. For each project i, it has a pure profit Pi and a minimum capital of Ci is needed to start the corresponding project. 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.

To sum up, pick a list of at most k distinct projects from given projects to maximize your final capital, and output your final maximized capital.

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.

Note:

You may assume all numbers in the input are non-negative integers.
The length of Profits array and Capital array will not exceed 50,000.
The answer is guaranteed to fit in a 32-bit signed integer.

Next Greater Element II

Given a circular array (the next element of the last element is the first element of the array), print the Next Greater Number for every element. 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, output -1 for this number.

Example 1:

Input: [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.

Note:
The length of given array won't exceed 10000.

Base 7

Given an integer, return its base 7 string representation.

Example 1:

Input: 100
Output: "202"

Example 2:

Input: -7
Output: "-10"

Note:
The input will be in range of [-1e7, 1e7].

Relative Ranks

Given scores of N athletes, find their relative ranks and the people with the top three highest scores, who will be awarded medals: "Gold Medal", "Silver Medal" and "Bronze Medal".

Example 1:

Input: [5, 4, 3, 2, 1]
Output: ["Gold Medal", "Silver Medal", "Bronze Medal", "4", "5"]
Explanation: The first three athletes got the top three highest scores, so they got "Gold Medal", "Silver Medal" and "Bronze Medal". For the left two athletes, you just need to output their relative ranks according to their scores.

Note:

N is a positive integer and won't exceed 10,000.
All the scores of athletes are guaranteed to be unique.

Perfect Number

We define the Perfect Number is a positive integer that is equal to the sum of all its positive divisors except itself.

Now, given an integer n, write a function that returns true when it is a perfect number and false when it is not.

Example:

Input: 28
Output: True
Explanation: 28 = 1 + 2 + 4 + 7 + 14

Note:
The input number n will not exceed 100,000,000. (1e8)

Most Frequent Subtree Sum

Given the root of a tree, you are asked to find the most frequent subtree sum. 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). So what is the most frequent subtree sum value? If there is a tie, return all the values with the highest frequency in any order.

Examples 1
Input:

```
  5
 /  \
2   -3
```

return [2, -3, 4], since all the values happen only once, return all of them in any order.

Examples 2
Input:

```
  5
 /  \
2   -5
```

return [2], since 2 happens twice, however -5 only occur once.

Note:
You may assume the sum of values in any subtree is in the range of 32-bit signed integer.

Find Bottom Left Tree Value

Given a binary tree, find the leftmost value in the last row of the tree.

Example 1:

Input:

```
    2
   / \
  1   3
```

Output:
1

Example 2:

Input:

Output:
7

Note:
You may assume the tree (i.e., the given root node) is not NULL.

Freedom Trail

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 in order to open the door.

Given a string ring, which represents the code engraved on the outer ring and another string key, which represents the keyword needs to be spelled. You need to find the minimum number of steps in order to spell all the characters in the keyword.

Initially, the first character of the ring is aligned at 12:00 direction. You need to spell all the characters in the string key one by one by rotating the ring clockwise or anticlockwise to make each character of the string key aligned at 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 one place, which counts as 1 step. The final purpose of the rotation is to align one of the string ring's characters at the 12:00 direction, where this character must equal to the character key[i].

If the character key[i] has been aligned at the 12:00 direction, you need to press the center button to spell, which also counts as 1

step. After the pressing, you could begin to spell the next character in the key (next stage), otherwise, you've finished all the spelling.

Example:

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.

Note:

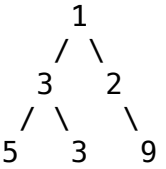
Length of both ring and key will be in range 1 to 100.
There are only lowercase letters in both strings and might be some duplcate characters in both strings.
It's guaranteed that string key could always be spelled by rotating the string ring.

Find Largest Value in Each Tree Row

You need to find the largest value in each row of a binary tree.

Example:

Input:



Output: [1, 3, 9]

Longest Palindromic Subsequence

Given a string s, find the longest palindromic subsequence's length in s. You may assume that the maximum length of s is 1000.

Example 1:
Input:

"bbbab"

Output:

4

One possible longest palindromic subsequence is "bbbb".

Example 2:
Input:

"cbbd"

Output:

2

One possible longest palindromic subsequence is "bb".

Super Washing Machines

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 representing the number of dresses in each washing machine from left to right on the line, you should find 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.

Example1

Input: [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
```

Example2

```
Input: [0,3,0]

Output: 2

Explanation:
1st move:    0 <-- 3     0    =>    1     2     0
2nd move:    1     2 --> 0    =>    1     1     1
```

Example3

```
Input: [0,2,0]

Output: -1

Explanation:
It's impossible to make all the three washing machines have the same
number of dresses.
```

Note:

The range of n is [1, 10000].
The range of dresses number in a super washing machine is [0, 1e5].

Detect Capital

Given a word, you need to judge whether the usage of capitals in it is right or not.

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 if it has more than one letter, like "Google".

Otherwise, we define that this word doesn't use capitals in a right way.

Example 1:

Input: "USA"
Output: True

Example 2:

Input: "FlaG"
Output: False

Note:
The input will be a non-empty word consisting of uppercase and lowercase latin letters.

Longest Uncommon Subsequence II

Given a list of strings, you need to find the longest uncommon subsequence among them. The longest uncommon subsequence is defined as the longest subsequence of one of these strings and this subsequence should not be any subsequence of the other strings.

A subsequence is a sequence that can be derived from one sequence by deleting some characters without changing the order of the remaining elements. Trivially, any string is a subsequence of itself and an empty string is a subsequence of any string.

The input will be a list of strings, and the output needs to be the length of the longest uncommon subsequence. If the longest uncommon subsequence doesn't exist, return -1.

Example 1:

Input: "aba", "cdc", "eae"
Output: 3

Note:

All the given strings' lengths will not exceed 10.
The length of the given list will be in the range of [2, 50].

Continuous Subarray Sum

Given a list of non-negative numbers and a target integer k, write a function to check if the array has a continuous subarray of size at least 2 that sums up to the multiple of k, that is, sums up to n*k where n is also an integer.

Example 1:

Input: [23, 2, 4, 6, 7], k=6
Output: True
Explanation: Because [2, 4] is a continuous subarray of size 2 and sums up to 6.

Example 2:

Input: [23, 2, 6, 4, 7], k=6
Output: True
Explanation: Because [23, 2, 6, 4, 7] is an continuous subarray of size 5 and sums up to 42.

Note:

The length of the array won't exceed 10,000.
You may assume the sum of all the numbers is in the range of a signed 32-bit integer.

Longest Word in Dictionary through Deleting

Given a string and a string dictionary, find the longest string in the dictionary that can be formed by deleting some characters of the given string. If there are more than one possible results, return the longest word with the smallest lexicographical order. If there is no possible result, return the empty string.

Example 1:

Input:
s = "abpcplea", d = ["ale","apple","monkey","plea"]

Output:
"apple"

Example 2:

Input:
s = "abpcplea", d = ["a","b","c"]

Output:
"a"

Note:

All the strings in the input will only contain lower-case letters.
The size of the dictionary won't exceed 1,000.
The length of all the strings in the input won't exceed 1,000.

Contiguous Array

Given a binary array, find the maximum length of a contiguous subarray with equal number of 0 and 1.

Example 1:

Input: [0,1]
Output: 2
Explanation: [0, 1] is the longest contiguous subarray with equal number of 0 and 1.

Example 2:

Input: [0,1,0]
Output: 2
Explanation: [0, 1] (or [1, 0]) is a longest contiguous subarray with equal number of 0 and 1.

Note:
The length of the given binary array will not exceed 50,000.

Beautiful Arrangement

Suppose you have N integers from 1 to N. We define a beautiful arrangement as an array that is constructed by these N numbers successfully if one of the following is true for the ith position (1 ≤ i ≤ N) in this array:

The number at the ith position is divisible by i.
i is divisible by the number at the ith position.

Now given N, how many beautiful arrangements can you construct?

Example 1:

Input: 2
Output: 2
Explanation:
The first beautiful arrangement is [1, 2]:
Number at the 1st position (i=1) is 1, and 1 is divisible by i (i=1).
Number at the 2nd position (i=2) is 2, and 2 is divisible by i (i=2).
The second beautiful arrangement is [2, 1]:
Number at the 1st position (i=1) is 2, and 2 is divisible by i (i=1).
Number at the 2nd position (i=2) is 1, and i (i=2) is divisible by 1.

Note:

N is a positive integer and will not exceed 15.

Minesweeper

Let's play the minesweeper game (Wikipedia, online game)!

You are given a 2D char matrix representing the game board. 'M' represents an unrevealed mine, 'E' represents an unrevealed empty

square, 'B' represents a revealed blank square that has no adjacent (above, below, left, right, and all 4 diagonals) mines, digit ('1' to '8') represents how many mines are adjacent to this revealed square, and finally 'X' represents a revealed mine.

Now given the next click position (row and column indices) 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 - change it to 'X'.
If an empty square ('E') with no adjacent mines is revealed, then change it to 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:

```
[['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']]
```

Explanation:

Example 2:

Input:

```
[['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']]
```

Explanation:

Note:

The range of the input matrix's height and width is [1,50].
The click position will only be an unrevealed square ('M' or 'E'), which also means the input board contains at least one clickable square.
The input board won't be a stage when game is over (some mines have been revealed).
For simplicity, not mentioned rules should be ignored in this problem. For example, you don't need to reveal all the unrevealed mines when the game is over, consider any cases that you will win the game or flag any squares.

Minimum Absolute Difference in BST

Given a binary search tree with non-negative values, find the minimum absolute difference between values of any two nodes.

Example:

Input:

```
   1
    \
     3
    /
   2
```

Output:
1

Explanation:
The minimum absolute difference is 1, which is the difference

between 2 and 1 (or between 2 and 3).

Note:
There are at least two nodes in this BST.

K-diff Pairs in an Array

Given an array of integers and an integer k, you need to find the number of unique k-diff pairs in the array. Here a k-diff pair is defined as an integer pair (i, j), where i and j are both numbers in the array and their absolute difference is k.

Example 1:

Input: [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:[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: [1, 3, 1, 5, 4], k = 0
Output: 1
Explanation: There is one 0-diff pair in the array, (1, 1).

Note:

The pairs (i, j) and (j, i) count as the same pair.
The length of the array won't exceed 10,000.
All the integers in the given input belong to the range: [-1e7, 1e7].

Encode and Decode TinyURL

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 the encode and decode methods for the TinyURL service. 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.

Complex Number Multiplication

Given two strings representing two complex numbers.

You need to return a string representing their multiplication. Note i2 = -1 according to the definition.

Example 1:

Input: "1+1i", "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: "1+-1i", "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.

Note:

The input strings will not have extra blank.
The input strings will be given in the form of a+bi, where the integer a and b will both belong to the range of [-100, 100]. And

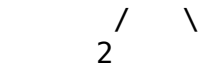
the output should be also in this form.

Convert BST to Greater Tree

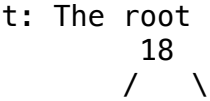
Given a Binary Search Tree (BST), convert it to a Greater Tree such that every key of the original BST is changed to the original key plus sum of all keys greater than the original key in BST.

Example:

Input: The root of a Binary Search Tree like this:



Output: The root of a Greater Tree like this:



Minimum Time Difference

Given a list of 24-hour clock time points in "Hour:Minutes" format, find the minimum minutes difference between any two time points in the list.

Example 1:

Input: ["23:59","00:00"]
Output: 1

Note:

The number of time points in the given list is at least 2 and won't exceed 20000.
The input time is legal and ranges from 00:00 to 23:59.

Reverse String II

Given a string and an integer k, you need to reverse the first k characters for every 2k characters counting from the start of the

string. If there are less 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 left the other as original.

Example:

Input: s = "abcdefg", k = 2
Output: "bacdfeg"

Restrictions:

The string consists of lower English letters only.
Length of the given string and k will in the range [1, 10000]

---
01 Matrix

Given a matrix consists of 0 and 1, find the distance of the nearest 0 for each cell.

The distance between two adjacent cells is 1.

Example 1:
Input:

0 0 0
0 1 0
0 0 0

Output:

0 0 0
0 1 0
0 0 0

Example 2:
Input:

0 0 0
0 1 0
1 1 1

Output:

0 0 0
0 1 0

1 2 1

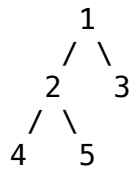
Note:

The number of elements of the given matrix will not exceed 10,000.
There are at least one 0 in the given matrix.
The cells are adjacent in only four directions: up, down, left and right.

Diameter of Binary Tree

Given a binary tree, you need to compute 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.

Example:
Given a binary tree



Return 3, which is the length of the path [4,2,1,3] or [5,2,1,3].

Note:
The length of path between two nodes is represented by the number of edges between them.

Remove Boxes

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 (composed of k boxes, k >= 1), remove them and get k*k points.

Find the maximum points you can get.

Example 1:
Input:

[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)

Note:
The number of boxes n would not exceed 100.

Friend Circles

There are N students in a class. Some of them are friends, while some are not. Their friendship is transitive in nature. For example, if A is a direct friend of B, and B is a direct friend of C, then A is an indirect friend of C. And we defined a friend circle is a group of students who are direct or indirect friends.

Given a N*N matrix M representing the friend relationship between students in the class. If M[i][j] = 1, then the ith and jth students are direct friends with each other, otherwise not. And you have to output the total number of friend circles among all the students.

Example 1:

Input:
[[1,1,0],
 [1,1,0],
 [0,0,1]]
Output: 2
Explanation:The 0th and 1st students are direct friends, so they are in a friend circle. The 2nd student himself is in a friend circle. So return 2.

Example 2:

Input:
[[1,1,0],
 [1,1,1],
 [0,1,1]]
Output: 1
Explanation:The 0th and 1st students are direct friends, the 1st and 2nd students are direct friends, so the 0th and 2nd students are indirect friends. All of them are in the same friend circle, so return 1.

Note:

N is in range [1,200].
M[i][i] = 1 for all students.
If M[i][j] = 1, then M[j][i] = 1.

Design TinyURL

Note: For the coding companion problem, please see: Encode and Decode TinyURL.

How would you design a URL shortening service that is similar to TinyURL?

Background:
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.

Requirements:

For instance, "http://tinyurl.com/4e9iAk" is the tiny url for the page "https://leetcode.com/problems/design-tinyurl". The identifier (the highlighted part) can be any string with 6 alphanumeric characters containing 0-9, a-z, A-Z.

Each shortened URL must be unique; that is, no two different URLs can be shortened to the same URL.

Note about Questions:Below are just a small subset of questions to get you started. In real world, there could be many follow ups and questions possible and the discussion is open-ended (No one true or correct way to solve a problem). If you have more ideas or questions, please ask in Discuss and we may compile it here!

Questions:

How many unique identifiers possible? Will you run out of unique URLs?
Should the identifier be increment or not? Which is easier to design? Pros and cons?
Mapping an identifier to an URL and its reversal - Does this problem ring a bell to you?
How do you store the URLs? Does a simple flat file database work?
What is the bottleneck of the system? Is it read-heavy or write-heavy?
Estimate the maximum number of URLs a single machine can store.
Estimate the maximum number of queries per second (QPS) for decoding a shortened URL in a single machine.
How would you scale the service? For example, a viral link which is shared in social media could result in a peak QPS at a moment's notice.
How could you handle redundancy? i,e, if a server is down, how could you ensure the service is still operational?
Keep URLs forever or prune, pros/cons? How we do pruning? (Contributed by @alex_svetkin)
What API would you provide to a third-party developer? (Contributed by @alex_svetkin)
If you can enable caching, what would you cache and what's the expiry time? (Contributed by @Humandroid)

```
.hilight {
  color: #d14;
  background-color: #f7f7f9;
  padding: 1px 3px;
  border: 1px solid #e1e1e8"
}
```


License Key Formatting

Now you are given a string S, which represents a software license key which we would like to format. The string S is composed of alphanumerical characters and dashes. The dashes split the alphanumerical characters within the string into groups. (i.e. if there are M dashes, the string is split into M+1 groups). The dashes in the given string are possibly misplaced.

We want each group of characters to be of length K (except for

possibly the first group, which could be shorter, but still must contain at least one character). To satisfy this requirement, we will reinsert dashes. Additionally, all the lower case letters in the string must be converted to upper case.

So, you are given a non-empty string S, representing a license key to format, and an integer K. And you need to return the license key formatted according to the description above.

Example 1:

Input: S = "2-4A0r7-4k", K = 4

Output: "24A0-R74K"

Explanation: The string S has been split into two parts, each part has 4 characters.

Example 2:

Input: S = "2-4A0r7-4k", K = 3

Output: "24-A0R-74K"

Explanation: The string S has been split into three parts, each part has 3 characters except the first part as it could be shorter as said above.

Note:

The length of string S will not exceed 12,000, and K is a positive integer.
String S consists only of alphanumerical characters (a-z and/or A-Z and/or 0-9) and dashes(-).
String S is non-empty.

Longest Absolute File Path

Suppose we abstract our file system by a string in the following manner:

The string "dir\n\tsubdir1\n\tsubdir2\n\t\tfile.ext" represents:

```
dir
    subdir1
```

```
    subdir2
        file.ext
```

The directory dir contains an empty sub-directory subdir1 and a sub-directory subdir2 containing a file file.ext.

The string "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext" represents:

```
dir
    subdir1
        file1.ext
        subsubdir1
    subdir2
        subsubdir2
            file2.ext
```

The directory dir contains two sub-directories subdir1 and subdir2. subdir1 contains a file file1.ext and an empty second-level sub-directory subsubdir1. subdir2 contains a second-level sub-directory subsubdir2 containing a file file2.ext.

We are interested in finding the longest (number of characters) absolute path to a file within our file system. For example, in the second example above, the longest absolute path is "dir/subdir2/subsubdir2/file2.ext", and its length is 32 (not including the double quotes).

Given a string representing the file system in the above format, return the length of the longest absolute path to file in the abstracted file system. If there is no file in the system, return 0.

Note:

The name of a file contains at least a . and an extension.
The name of a directory or sub-directory will not contain a ..

Time complexity required: O(n) where n is the size of the input string.

Notice that a/aa/aaa/file1.txt is not the longest file path, if there is another path aaaaaaaaaaaaaaaaaaaa/sth.png.

Encode and Decode TinyURL

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 the encode and decode methods for the TinyURL service. 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.

Coin Change 2

You are given coins of different denominations and a total amount of money. Write a function to compute the number of combinations that make up that amount. You may assume that you have infinite number of each kind of coin.

Note:
You can assume that

- 0 <= amount <= 5000
- 1 <= coin <= 5000
- the number of coins is less than 500
- the answer is guaranteed to fit into 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
```

Poor Pigs

There are 1000 buckets, one and only one of them contains poison, the rest are filled with water. They all look the same. If a pig drinks that poison it will die within 15 minutes. What is the minimum amount of pigs you need to figure out which bucket contains the poison within one hour.

Answer this question, and write an algorithm for the follow-up general case.

Follow-up:

If there are n buckets and a pig drinking poison will die within m minutes, how many pigs (x) you need to figure out the "poison" bucket within p minutes? There is exact one bucket with poison.

Minimum Genetic Mutation

A gene string can be represented by an 8-character long string, with choices from "A", "C", "G", "T".

Suppose we need to investigate about a mutation (mutation from "start" to "end"), where ONE mutation is defined as ONE single character changed in the gene string.

For example, "AACCGGTT" -> "AACCGGTA" is 1 mutation.

Also, there is a given gene "bank", which records all the valid gene mutations. A gene must be in the bank to make it a valid gene string.

Now, given 3 things - start, end, bank, your task is to determine what is the minimum number of mutations needed to mutate from "start" to "end". If there is no such a mutation, return -1.

Note:

Starting point is assumed to be valid, so it might not be included in the bank.
If multiple mutations are needed, all mutations during in the sequence must be valid.
You may assume start and end string is not the same.

Example 1:

start: "AACCGGTT"
end: "AACCGGTA"
bank: ["AACCGGTA"]

return: 1

Example 2:

start: "AACCGGTT"
end: "AAACGGTA"
bank: ["AACCGGTA", "AACCGCTA", "AAACGGTA"]

return: 2

Example 3:

start: "AAAAACCC"
end: "AACCCCCC"
bank: ["AAAACCCC", "AAACCCCC", "AACCCCCC"]

return: 3

LFU Cache

Design and implement a data structure for Least Frequently Used (LFU) cache. It should support the following operations: get and put.

get(key) - Get the value (will always be positive) of the key if the key exists in the cache, otherwise return -1.
put(key, value) - Set or insert the value if the key is not already present. When the cache reaches its capacity, it should invalidate the least frequently used item before inserting a new item. For the purpose of this problem, when there is a tie (i.e., two or more keys that have the same frequency), the least recently used key would be evicted.

Follow up:
Could you do both operations in O(1) time complexity?

Example:

```
LFUCache cache = new LFUCache( 2 /* capacity */ );

cache.put(1, 1);
cache.put(2, 2);
cache.get(1);       // returns 1
cache.put(3, 3);    // evicts key 2
cache.get(2);       // returns -1 (not found)
cache.get(3);       // returns 3.
cache.put(4, 4);    // evicts key 1.
cache.get(1);       // returns -1 (not found)
cache.get(3);       // returns 3
cache.get(4);       // returns 4
```


Longest Palindromic Subsequence

Given a string s, find the longest palindromic subsequence's length in s. You may assume that the maximum length of s is 1000.

Example 1:
Input:

"bbbab"

Output:

4

One possible longest palindromic subsequence is "bbbb".

Example 2:
Input:

"cbbd"

Output:

2

One possible longest palindromic subsequence is "bb".