# 1.Two Sum(easy)

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**].

输入：

[3,2,4]

6

输出：

[1,2]

# 2. Add Two Numbers(medium)

Medium

65061693FavoriteShare

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.

**Example:**

**Input:** (2 -> 4 -> 3) + (5 -> 6 -> 4)

**Output:** 7 -> 0 -> 8

**Explanation:** 342 + 465 = 807.

# 3. Longest Substring Without Repeating Characters(medium)

Medium

7087418FavoriteShare

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

**Example 1:**

**Input:** "abcabcbb"

**Output:** 3

**Explanation:** The answer is "abc", with the length of 3.

**Example 2:**

**Input:** "bbbbb"

**Output:** 1

**Explanation:** The answer is "b", with the length of 1.

**Example 3:**

**Input:** "pwwkew"

**Output:** 3

**Explanation:** 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.

# 4. Median of Two Sorted Arrays(hard)

Hard

5490812FavoriteShare

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)).

You may assume **nums1** and **nums2** cannot be both empty.

**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

# 5. Longest Palindromic Substring(medium)

Medium

4908434FavoriteShare

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

**Example 1:**

**Input:** "babad"

**Output:** "bab"

**Note:** "aba" is also a valid answer.

**Example 2:**

**Input:** "cbbd"

**Output:** "bb"

# 7. Reverse Integer(easy)

Given a 32-bit signed integer, reverse digits of an integer.

**Example 1:**

**Input:** 123

**Output:** 321

**Example 2:**

**Input:** -123

**Output:** -321

**Example 3:**

**Input:** 120

**Output:** 21

**Note:**  
Assume we are dealing with an environment which could only hold integers within the 32-bit signed integer range. For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

# 9. Palindrome Number(easy)

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

**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.

# 13. Roman to Integer(easy)

Easy

10242331FavoriteShare

Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

**Symbol** **Value**

I 1

V 5

X 10

L 50

C 100

D 500

M 1000

For example, two is written as II in Roman numeral, just two one's added together. Twelve is written as, XII, which is simply X + II. The number twenty seven is written as XXVII, which is XX + V + II.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not IIII. Instead, the number four is written as IV. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as IX. There are six instances where subtraction is used:

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

Given a roman numeral, convert it to an integer. Input is guaranteed to be within the range from 1 to 3999.

**Example 1:**

**Input:** "III"

**Output:** 3

**Example 2:**

**Input:** "IV"

**Output:** 4

**Example 3:**

**Input:** "IX"

**Output:** 9

**Example 4:**

**Input:** "LVIII"

**Output:** 58

**Explanation:** L = 50, V= 5, III = 3.

**Example 5:**

**Input:** "MCMXCIV"

**Output:** 1994

**Explanation:** M = 1000, CM = 900, XC = 90 and IV = 4.

# 14. Longest Common Prefix(easy)

Easy

10391106FavoriteShare

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

If there is no common prefix, return an empty string "".

**Example 1:**

**Input:** ["flower","flow","flight"]

**Output:** "fl"

**Example 2:**

**Input:** ["dog","racecar","car"]

**Output:** ""

**Explanation:** There is no common prefix among the input strings.

**Note:**

All given inputs are in lowercase letters a-z.

# 20. Valid Parentheses(easy)

Easy

2371117FavoriteShare

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

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

**Example 1:**

**Input:** "()"

**Output:** true

**Example 2:**

**Input:** "()[]{}"

**Output:** true

**Example 3:**

**Input:** "(]"

**Output:** false

**Example 4:**

**Input:** "([)]"

**Output:** false

**Example 5:**

**Input:** "{[]}"

**Output:** true

# 21. Merge Two Sorted Lists(easy)

Easy

1753240FavoriteShare

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.

**Example:**

**Input:** 1->2->4, 1->3->4

**Output:** 1->1->2->3->4->4

# 26. Remove Duplicates from Sorted Array(easy)

Easy

12682647FavoriteShare

Given a sorted array nums, remove the duplicates [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) such that each element appear only once and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

**Example 1:**

Given 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 returned length.

**Example 2:**

Given nums = **[0,0,1,1,1,2,2,3,3,4]**,

Your function should return length = **5**, with the first five elements of *nums* being modified to **0**, **1**, **2**, **3**, and **4** respectively.

It doesn't matter what values are set beyond the returned length.

**Clarification:**

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by **reference**, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

// **nums** is passed in by reference. (i.e., without making a copy)

int len = removeDuplicates(nums);

// any modification to **nums** in your function would be known by the caller.

// using the length returned by your function, it prints the first **len** elements.

for (int i = 0; i < len; i++) {

    print(nums[i]);

}

# 27. Remove Element(easy)

Easy

6751466FavoriteShare

Given an array nums and a value val, remove all instances of that value [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

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

**Example 1:**

Given nums = **[3,2,2,3]**, val = **3**,

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

It doesn't matter what you leave beyond the returned length.

**Example 2:**

Given nums = **[0,1,2,2,3,0,4,2]**, val = **2**,

Your function should return length = **5**, with the first five elements of *nums* containing **0**, **1**, **3**, **0**, and **4**.

Note that the order of those five elements can be arbitrary.

It doesn't matter what values are set beyond the returned length.

**Clarification:**

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by **reference**, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

// **nums** is passed in by reference. (i.e., without making a copy)

int len = removeElement(nums, val);

// any modification to **nums** in your function would be known by the caller.

// using the length returned by your function, it prints the first **len** elements.

for (int i = 0; i < len; i++) {

    print(nums[i]);

}

# 28. Implement strStr()(easy)

Easy

7241117FavoriteShare

Implement [strStr()](http://www.cplusplus.com/reference/cstring/strstr/).

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

**Example 1:**

**Input:** haystack = "hello", needle = "ll"

**Output:** 2

**Example 2:**

**Input:** haystack = "aaaaa", needle = "bba"

**Output:** -1

**Clarification:**

What should we return when needle is an empty string? This is a great question to ask during an interview.

For the purpose of this problem, we will return 0 when needle is an empty string. This is consistent to C's [strstr()](http://www.cplusplus.com/reference/cstring/strstr/) and Java's [indexOf()](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#indexOf(java.lang.String)).

Accepted

361,703

Submissions

1,172,841

# 35. Search Insert Position(easy)

Easy

1068154FavoriteShare

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.

**Example 1:**

**Input:** [1,3,5,6], 5

**Output:** 2

**Example 2:**

**Input:** [1,3,5,6], 2

**Output:** 1

**Example 3:**

**Input:** [1,3,5,6], 7

**Output:** 4

**Example 4:**

**Input:** [1,3,5,6], 0

**Output:** 0

# 38. Count and Say(easy)

Easy

6254420FavoriteShare

The count-and-say sequence is the sequence of integers with the first five terms as following:

1. 1

2. 11

3. 21

4. 1211

5. 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* where 1 ≤ *n* ≤ 30, generate the *n*th term of the count-and-say sequence.

Note: Each term of the sequence of integers will be represented as a string.

**Example 1:**

**Input:** 1

**Output:** "1"

**Example 2:**

**Input:** 4

**Output:** "1211"

# 53. Maximum Subarray(easy)

Easy

3394119FavoriteShare

Given an integer array nums, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

**Example:**

**Input:** [-2,1,-3,4,-1,2,1,-5,4],

**Output:** 6

**Explanation:** [4,-1,2,1] has the largest sum = 6.

**Follow up:**

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

就是一个最大子列和问题，用动态规划的思路解决~

# 58. Length of Last Word(easy)

Easy

3071237FavoriteShare

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.

**Example:**

**Input:** "Hello World"

**Output:** 5

# 66. Plus One(easy)

Easy

7141303FavoriteShare

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

The digits are stored such that the most significant digit is at the head of the list, and each element in the array contain a single digit.

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

**Example 1:**

**Input:** [1,2,3]

**Output:** [1,2,4]

**Explanation:** The array represents the integer 123.

**Example 2:**

**Input:** [4,3,2,1]

**Output:** [4,3,2,2]

**Explanation:** The array represents the integer 4321.

# 67. Add Binary(easy)

Easy

784163FavoriteShare

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

The input strings are both **non-empty** and contains only characters 1 or 0.

**Example 1:**

**Input:** a = "11", b = "1"

**Output:** "100"

**Example 2:**

**Input:** a = "1010", b = "1011"

**Output:** "10101"

# 69. Sqrt(x)(easy)

Easy

6231151FavoriteShare

Implement int sqrt(int x).

Compute and return the square root of *x*, where *x* is guaranteed to be a non-negative integer.

Since the return type is an integer, the decimal digits are truncated and only the integer part of the result is returned.

**Example 1:**

**Input:** 4

**Output:** 2

**Example 2:**

**Input:** 8

**Output:** 2

**Explanation:** The square root of 8 is 2.82842..., and since

  the decimal part is truncated, 2 is returned.

注意int类型的表示范围可能导致溢出

# 70. Climbing Stairs(easy)

Easy

177167FavoriteShare

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.

**Example 1:**

**Input:** 2

**Output:** 2

**Explanation:** There are two ways to climb to the top.

1. 1 step + 1 step

2. 2 steps

**Example 2:**

**Input:** 3

**Output:** 3

**Explanation:** There are three ways to climb to the top.

1. 1 step + 1 step + 1 step

2. 1 step + 2 steps

3. 2 steps + 1 step

本质就是个斐波那契

# 83. Remove Duplicates from Sorted List(easy)

Easy

65772FavoriteShare

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

**Example 1:**

**Input:** 1->1->2

**Output:** 1->2

**Example 2:**

**Input:** 1->1->2->3->3

**Output:** 1->2->3

# 88. Merge Sorted Array(easy)

Easy

9392470FavoriteShare

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

**Note:**

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

**Example:**

**Input:**

nums1 = [1,2,3,0,0,0], m = 3

nums2 = [2,5,6], n = 3

**Output:** [1,2,2,3,5,6]

# 100. Same Tree(easy)

Easy

99931FavoriteShare

Given two binary trees, write a function to check if they are the same or not.

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

**Example 1:**

**Input:** 1 1

/ \ / \

2 3 2 3

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

**Output:** true

**Example 2:**

**Input:** 1 1

/ \

2 2

[1,2], [1,null,2]

**Output:** false

**Example 3:**

**Input:** 1 1

/ \ / \

2 1 1 2

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

**Output:** false

# 101. Symmetric Tree(easy)

Easy

190843FavoriteShare

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.

# 104. Maximum Depth of Binary Tree(easy)

Easy

113247FavoriteShare

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.

**Note:** A leaf is a node with no children.

**Example:**

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

3

/ \

9 20

/ \

15 7

return its depth = 3.

# 107. Binary Tree Level Order Traversal II(easy)

Easy

639113FavoriteShare

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],

3

/ \

9 20

/ \

15 7

return its bottom-up level order traversal as:

[

[15,7],

[9,20],

[3]

]

# 108. Convert Sorted Array to Binary Search Tree(easy)

Easy

97599FavoriteShare

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

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.

**Example:**

Given the sorted array: [-10,-3,0,5,9],

One possible answer is: [0,-3,9,-10,null,5], which represents the following height balanced BST:

0

/ \

-3 9

/ /

-10 5

# 110. Balanced Binary Tree(easy)

Easy

106492FavoriteShare

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.

**Example 1:**

Given the following tree [3,9,20,null,null,15,7]:

3

/ \

9 20

/ \

15 7

Return true.  
  
**Example 2:**

Given the following tree [1,2,2,3,3,null,null,4,4]:

1

/ \

2 2

/ \

3 3

/ \

4 4

Return false.

# 111. Minimum Depth of Binary Tree(easy)

Easy

644322FavoriteShare

Given a binary tree, find its minimum depth.

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

**Note:** A leaf is a node with no children.

**Example:**

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

3

/ \

9 20

/ \

15 7

return its minimum depth = 2.

# 112. Path Sum(easy)

Easy

806261FavoriteShare

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.

**Note:** A leaf is a node with no children.

**Example:**

Given the below binary tree and sum = 22,

**5**

**/** \

**4** 8

**/** / \

**11** 13 4

/ **\** \

7 **2** 1

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

# 118. Pascal's Triangle(easy)

Easy

62072FavoriteShare

Given a non-negative integer *numRows*, generate the first *numRows* of Pascal's triangle.

  
In Pascal's triangle, each number is the sum of the two numbers directly above it.

**Example:**

**Input:** 5

**Output:**

[

[1],

[1,1],

[1,2,1],

[1,3,3,1],

[1,4,6,4,1]

]

# 119. Pascal's Triangle II(easy)

Easy

419153FavoriteShare

Given a non-negative index *k* where *k* ≤ 33, return the *k*th index row of the Pascal's triangle.

Note that the row index starts from 0.

  
In Pascal's triangle, each number is the sum of the two numbers directly above it.

**Example:**

**Input:** 3

**Output:** [1,3,3,1]

**Follow up:**

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

# 121. Best Time to Buy and Sell Stock(easy)

Easy

2273113FavoriteShare

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

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

Note that you cannot sell a stock before you buy one.

**Example 1:**

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

**Output:** 5

**Explanation:** Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = 6-1 = 5.

  Not 7-1 = 6, as selling price needs to be larger than buying price.

**Example 2:**

**Input:** [7,6,4,3,1]

**Output:** 0

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

# 122. Best Time to Buy and Sell Stock II(easy)

Easy

8631211FavoriteShare

Say you have an array for which the *i*th 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 (i.e., buy one and sell one share of the stock multiple times).

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

**Example 1:**

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

**Output:** 7

**Explanation:** Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4.

  Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3.

**Example 2:**

**Input:** [1,2,3,4,5]

**Output:** 4

**Explanation:** Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4.

  Note that you cannot buy on day 1, buy on day 2 and sell them later, as you are

  engaging multiple transactions at the same time. You must sell before buying again.

**Example 3:**

**Input:** [7,6,4,3,1]

**Output:** 0

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

# 125. Valid Palindrome(easy)

Easy

5401519FavoriteShare

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

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

**Example 1:**

**Input:** "A man, a plan, a canal: Panama"

**Output:** true

**Example 2:**

**Input:** "race a car"

**Output:** false

# 136. Single Number(easy)

Easy

221183FavoriteShare

Given a **non-empty** 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?

**Example 1:**

**Input:** [2,2,1]

**Output:** 1

**Example 2:**

**Input:** [4,1,2,1,2]

**Output:** 4

# 141. Linked List Cycle(easy)

Easy

1410139FavoriteShare

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

To represent a cycle in the given linked list, we use an integer pos which represents the position (0-indexed) in the linked list where tail connects to. If pos is -1, then there is no cycle in the linked list.

**Example 1:**

**Input:** head = [3,2,0,-4], pos = 1

**Output:** true

**Explanation:** There is a cycle in the linked list, where tail connects to the second node.



**Example 2:**

**Input:** head = [1,2], pos = 0

**Output:** true

**Explanation:** There is a cycle in the linked list, where tail connects to the first node.



**Example 3:**

**Input:** head = [1], pos = -1

**Output:** false

**Explanation:** There is no cycle in the linked list.



**Follow up:**

Can you solve it using *O(1)* (i.e. constant) memory?

# 155. Min Stack(easy)

Easy

1647172FavoriteShare

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.

# 167. Two Sum II - Input array is sorted(easy)

Easy

839356FavoriteShare

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.

**Note:**

* 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.

**Example:**

**Input:** numbers = [2,7,11,15], target = 9

**Output:** [1,2]

**Explanation:** The sum of 2 and 7 is 9. Therefore index1 = 1, index2 = 2.

# 168. Excel Sheet Column Title(easy)

Easy

678132FavoriteShare

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

...

**Example 1:**

**Input:** 1

**Output:** "A"

**Example 2:**

**Input:** 28

**Output:** "AB"

**Example 3:**

**Input:** 701

**Output:** "ZY"

# 169. Majority Element(easy)

Easy

1545136FavoriteShare

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.

**Example 1:**

**Input:** [3,2,3]

**Output:** 3

**Example 2:**

**Input:** [2,2,1,1,1,2,2]

**Output:** 2

# 171. Excel Sheet Column Number(easy)

Easy

51693FavoriteShare

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

...

**Example 1:**

**Input:** "A"

**Output:** 1

**Example 2:**

**Input:** "AB"

**Output:** 28

**Example 3:**

**Input:** "ZY"

**Output:** 701

# 172. Factorial Trailing Zeroes(easy)

Easy

459657FavoriteShare

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

**Example 1:**

**Input:** 3

**Output:** 0

**Explanation:** 3! = 6, no trailing zero.

**Example 2:**

**Input:** 5

**Output:** 1

**Explanation:** 5! = 120, one trailing zero.

**Note:**Your solution should be in logarithmic time complexity.

# 175. Combine Two Tables(easy)

Easy

67295FavoriteShare

SQL Schema

Table: Person

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| PersonId | int |

| FirstName | varchar |

| LastName | varchar |

+-------------+---------+

PersonId is the primary key column for this table.

Table: Address

+-------------+---------+

| Column Name | Type |

+-------------+---------+

| AddressId | int |

| PersonId | int |

| City | varchar |

| State | varchar |

+-------------+---------+

AddressId is the primary key column for this table.

Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people:

FirstName, LastName, City, State

# 176. Second Highest Salary(easy)

Easy

518270FavoriteShare

SQL Schema

Write a SQL query to get the second highest salary from the Employee table.

+----+--------+

| Id | Salary |

+----+--------+

| 1 | 100 |

| 2 | 200 |

| 3 | 300 |

+----+--------+

For example, given the above Employee table, the query should return 200 as the second highest salary. If there is no second highest salary, then the query should return null.

+---------------------+

| SecondHighestSalary |

+---------------------+

| 200 |

+---------------------+

# 181. Employees Earning More Than Their Managers(easy)

Easy

37142FavoriteShare

SQL Schema

The Employee table holds all employees including their managers. Every employee has an Id, and there is also a column for the manager Id.

+----+-------+--------+-----------+

| Id | Name | Salary | ManagerId |

+----+-------+--------+-----------+

| 1 | Joe | 70000 | 3 |

| 2 | Henry | 80000 | 4 |

| 3 | Sam | 60000 | NULL |

| 4 | Max | 90000 | NULL |

+----+-------+--------+-----------+

Given the Employee table, write a SQL query that finds out employees who earn more than their managers. For the above table, Joe is the only employee who earns more than his manager.

+----------+

| Employee |

+----------+

| Joe |

+----------+

# 182. Duplicate Emails(easy)

Easy

29918FavoriteShare

SQL Schema

Write a SQL query to find all duplicate emails in a table named Person.

+----+---------+

| Id | Email |

+----+---------+

| 1 | a@b.com |

| 2 | c@d.com |

| 3 | a@b.com |

+----+---------+

For example, your query should return the following for the above table:

+---------+

| Email |

+---------+

| a@b.com |

+---------+

**Note**: All emails are in lowercase.

# 183. Customers Who Never Order(easy)

Easy

25632FavoriteShare

SQL Schema

Suppose that a website contains two tables, the Customers table and the Orders table. Write a SQL query to find all customers who never order anything.

Table: Customers.

+----+-------+

| Id | Name |

+----+-------+

| 1 | Joe |

| 2 | Henry |

| 3 | Sam |

| 4 | Max |

+----+-------+

Table: Orders.

+----+------------+

| Id | CustomerId |

+----+------------+

| 1 | 3 |

| 2 | 1 |

+----+------------+

Using the above tables as example, return the following:

+-----------+

| Customers |

+-----------+

| Henry |

| Max |

+-----------+

# 189. Rotate Array(easy)

Easy

1443619FavoriteShare

Given an array, rotate the array to the right by *k* steps, where *k* is non-negative.

**Example 1:**

**Input:** [1,2,3,4,5,6,7] and *k* = 3

**Output:** [5,6,7,1,2,3,4]

**Explanation:**

rotate 1 steps to the right: [7,1,2,3,4,5,6]

rotate 2 steps to the right: [6,7,1,2,3,4,5]

rotate 3 steps to the right: [5,6,7,1,2,3,4]

**Example 2:**

**Input:** [-1,-100,3,99] and *k* = 2

**Output:** [3,99,-1,-100]

**Explanation:**

rotate 1 steps to the right: [99,-1,-100,3]

rotate 2 steps to the right: [3,99,-1,-100]

**Note:**

* Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.
* Could you do it in-place with O(1) extra space?

# 190. Reverse Bits(easy)

Easy

589197FavoriteShare

Reverse bits of a given 32 bits unsigned integer.

**Example 1:**

**Input:** 00000010100101000001111010011100

**Output:** 00111001011110000010100101000000

**Explanation:** The input binary string **00000010100101000001111010011100** represents the unsigned integer 43261596, so return 964176192 which its binary representation is **00111001011110000010100101000000**.

**Example 2:**

**Input:** 11111111111111111111111111111101

**Output:** 10111111111111111111111111111111

**Explanation:** The input binary string **11111111111111111111111111111101** represents the unsigned integer 4294967293, so return 3221225471 which its binary representation is **10101111110010110010011101101001**.

**Note:**

* Note that in some languages such as Java, there is no unsigned integer type. In this case, both input and output will be given as signed integer type and should not affect your implementation, as the internal binary representation of the integer is the same whether it is signed or unsigned.
* In Java, the compiler represents the signed integers using [2's complement notation](https://en.wikipedia.org/wiki/Two%27s_complement). Therefore, in **Example 2** above the input represents the signed integer -3 and the output represents the signed integer -1073741825.

**Follow up**:

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

# 191. Number of 1 Bits(easy)

Easy

475402FavoriteShare

Write a function that takes an unsigned integer and return the number of '1' bits it has (also known as the [Hamming weight](http://en.wikipedia.org/wiki/Hamming_weight)).

**Example 1:**

**Input:** 00000000000000000000000000001011

**Output:** 3

**Explanation:** The input binary string **00000000000000000000000000001011** has a total of three '1' bits.

**Example 2:**

**Input:** 00000000000000000000000010000000

**Output:** 1

**Explanation:** The input binary string **00000000000000000000000010000000** has a total of one '1' bit.

**Example 3:**

**Input:** 11111111111111111111111111111101

**Output:** 31

**Explanation:** The input binary string **11111111111111111111111111111101** has a total of thirty one '1' bits.

**Note:**

* Note that in some languages such as Java, there is no unsigned integer type. In this case, the input will be given as signed integer type and should not affect your implementation, as the internal binary representation of the integer is the same whether it is signed or unsigned.
* In Java, the compiler represents the signed integers using [2's complement notation](https://en.wikipedia.org/wiki/Two%27s_complement). Therefore, in **Example 3** above the input represents the signed integer -3.

**Follow up**:

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

# 193. Valid Phone Numbers(easy)

Easy

101247FavoriteShare

Given a text file file.txt that contains list of phone numbers (one per line), write a one liner bash script to print all valid phone numbers.

You may assume that a valid phone number must appear in one of the following two formats: (xxx) xxx-xxxx or xxx-xxx-xxxx. (x means a digit)

You may also assume each line in the text file must not contain leading or trailing white spaces.

**Example:**

Assume that file.txt has the following content:

987-123-4567

123 456 7890

(123) 456-7890

Your script should output the following valid phone numbers:

987-123-4567

(123) 456-7890

# 195. Tenth Line(easy)

Easy

127112FavoriteShare

Given a text file file.txt, print just the 10th line of the file.

**Example:**

Assume that file.txt has the following content:

Line 1

Line 2

Line 3

Line 4

Line 5

Line 6

Line 7

Line 8

Line 9

Line 10

Your script should output the tenth line, which is:

Line 10

**Note:**  
1. If the file contains less than 10 lines, what should you output?  
2. There's at least three different solutions. Try to explore all possibilities.

# 196. Delete Duplicate Emails(easy)

Easy

289351FavoriteShare

Write a SQL query to **delete** all duplicate email entries in a table named Person, keeping only unique emails based on its *smallest* **Id**.

+----+------------------+

| Id | Email |

+----+------------------+

| 1 | john@example.com |

| 2 | bob@example.com |

| 3 | john@example.com |

+----+------------------+

Id is the primary key column for this table.

For example, after running your query, the above Person table should have the following rows:

+----+------------------+

| Id | Email |

+----+------------------+

| 1 | john@example.com |

| 2 | bob@example.com |

+----+------------------+

**Note:**

Your output is the whole Person table after executing your sql. Use delete statement.

# 197. Rising Temperature(easy)

Easy

263109FavoriteShare

SQL Schema

Given a Weather table, write a SQL query to find all dates' Ids with higher temperature compared to its previous (yesterday's) dates.

+---------+------------------+------------------+

| Id(INT) | RecordDate(DATE) | Temperature(INT) |

+---------+------------------+------------------+

| 1 | 2015-01-01 | 10 |

| 2 | 2015-01-02 | 25 |

| 3 | 2015-01-03 | 20 |

| 4 | 2015-01-04 | 30 |

+---------+------------------+------------------+

For example, return the following Ids for the above Weather table:

+----+

| Id |

+----+

| 2 |

| 4 |

+----+

# 198. House Robber(easy)

Easy

279987FavoriteShare

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**.

**Example 1:**

**Input:** [1,2,3,1]

**Output:** 4

**Explanation:** Rob house 1 (money = 1) and then rob house 3 (money = 3).

  Total amount you can rob = 1 + 3 = 4.

**Example 2:**

**Input:** [2,7,9,3,1]

**Output:** 12

**Explanation:** Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money = 1).

  Total amount you can rob = 2 + 9 + 1 = 12.

# 202. Happy Number(easy)

Easy

952248FavoriteShare

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:**

**Input:** 19

**Output:** true

**Explanation:**

12 + 92 = 82

82 + 22 = 68

62 + 82 = 100

12 + 02 + 02 = 1

# 203. Remove Linked List Elements(easy)

Easy

91557FavoriteShare

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

**Example:**

**Input:** 1->2->6->3->4->5->6, ***val*** = 6

**Output:** 1->2->3->4->5

# 204. Count Primes(easy)

Easy

1203432FavoriteShare

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

**Example:**

**Input:** 10

**Output:** 4

**Explanation:** There are 4 prime numbers less than 10, they are 2, 3, 5, 7.

# 205. Isomorphic Strings(easy)

Easy

829240FavoriteShare

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.

**Example 1:**

**Input:** ***s*** = "egg", ***t =*** "add"

**Output:** true

**Example 2:**

**Input:** ***s*** = "foo", ***t =*** "bar"

**Output:** false

**Example 3:**

**Input:** ***s*** = "paper", ***t =*** "title"

**Output:** true

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

# 206. Reverse Linked List(easy)

Easy

265169FavoriteShare

Reverse a singly linked list.

**Example:**

**Input:** 1->2->3->4->5->NULL

**Output:** 5->4->3->2->1->NULL

**Follow up:**

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

# 217. Contains Duplicate(easy)

Easy

452558FavoriteShare

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.

**Example 1:**

**Input:** [1,2,3,1]

**Output:** true

**Example 2:**

**Input:** [1,2,3,4]

**Output:** false

**Example 3:**

**Input:** [1,1,1,3,3,4,3,2,4,2]

**Output:** true

# 219. Contains Duplicate II(easy)

Easy

553678FavoriteShare

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*.

**Example 1:**

**Input:** nums = [1,2,3,1], k = 3

**Output:** true

**Example 2:**

**Input:** nums = [1,0,1,1], k = 1

**Output:** true

**Example 3:**

**Input:** nums = [1,2,3,1,2,3], k = 2

**Output:** false

# 225. Implement Stack using Queues(easy)

Easy

373454FavoriteShare

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.

**Example:**

MyStack stack = new MyStack();

stack.push(1);

stack.push(2);

stack.top(); // returns 2

stack.pop(); // returns 2

stack.empty(); // returns false

**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).

# 226. Invert Binary Tree(easy)

Easy

193432FavoriteShare

Invert a binary tree.

**Example:**

Input:

4

/ \

2 7

/ \ / \

1 3 6 9

Output:

4

/ \

7 2

/ \ / \

9 6 3 1

**Trivia:**  
This problem was inspired by [this original tweet](https://twitter.com/mxcl/status/608682016205344768) by [Max Howell](https://twitter.com/mxcl):

Google: 90% of our engineers use the software you wrote (Homebrew), but you can’t invert a binary tree on a whiteboard so f\*\*\* off.

# 231. Power of Two(easy)

Easy

487136FavoriteShare

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

**Example 1:**

**Input:** 1

**Output:** true

**Explanation:** 20 = 1

**Example 2:**

**Input:** 16

**Output:** true

**Explanation:** 24 = 16

**Example 3:**

**Input:** 218

**Output:** false

# 232. Implement Queue using Stacks(easy)

Easy

638114FavoriteShare

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.

**Example:**

MyQueue queue = new MyQueue();

queue.push(1);

queue.push(2);

queue.peek(); // returns 1

queue.pop(); // returns 1

queue.empty(); // returns false

**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).

# 235. Lowest Common Ancestor of a Binary Search Tree(easy)

Easy

122288FavoriteShare

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](https://en.wikipedia.org/wiki/Lowest_common_ancestor): “The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow **a node to be a descendant of itself**).”

Given binary search tree:  root = [6,2,8,0,4,7,9,null,null,3,5]



**Example 1:**

**Input:** root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8

**Output:** 6

**Explanation:** The LCA of nodes 2 and 8 is 6.

**Example 2:**

**Input:** root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4

**Output:** 2

**Explanation:** The LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

**Note:**

* All of the nodes' values will be unique.
* p and q are different and both values will exist in the BST.

# 237. Delete Node in a Linked List(easy)

Easy

8544087FavoriteShare

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

Given linked list -- head = [4,5,1,9], which looks like following:



**Example 1:**

**Input:** head = [4,5,1,9], node = 5

**Output:** [4,1,9]

**Explanation:** You are given the second node with value 5, the linked list should become 4 -> 1 -> 9 after calling your function.

**Example 2:**

**Input:** head = [4,5,1,9], node = 1

**Output:** [4,5,9]

**Explanation:** You are given the third node with value 1, the linked list should become 4 -> 5 -> 9 after calling your function.

**Note:**

* The linked list will have at least two elements.
* All of the nodes' values will be unique.
* The given node will not be the tail and it will always be a valid node of the linked list.
* Do not return anything from your function.

# 242. Valid Anagram(easy)

Easy

814116FavoriteShare

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

**Example 1:**

**Input:** *s* = "anagram", *t* = "nagaram"

**Output:** true

**Example 2:**

**Input:** *s* = "rat", *t* = "car"

**Output:** 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?

# 257. Binary Tree Paths(easy)

Easy

99576FavoriteShare

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

**Note:** A leaf is a node with no children.

**Example:**

**Input:**

1

/ \

2 3

\

5

**Output:** ["1->2->5", "1->3"]

**Explanation:** All root-to-leaf paths are: 1->2->5, 1->3

# 258. Add Digits(easy)

Easy

503837FavoriteShare

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

**Example:**

**Input:** 38

**Output:** 2

**Explanation:** 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?

# 263. Ugly Number(easy)

Easy

264479FavoriteShare

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.

**Example 1:**

**Input:** 6

**Output:** true

**Explanation:** 6 = 2 × 3

**Example 2:**

**Input:** 8

**Output:** true

**Explanation:** 8 = 2 × 2 × 2

**Example 3:**

**Input:** 14

**Output:** false

**Explanation:** 14 is not ugly since it includes another prime factor 7.

**Note:**

1. 1 is typically treated as an ugly number.
2. Input is within the 32-bit signed integer range: [−231,  231− 1].

# 268. Missing Number(easy)

Easy

10711462FavoriteShare

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

**Example 1:**

**Input:** [3,0,1]

**Output:** 2

**Example 2:**

**Input:** [9,6,4,2,3,5,7,0,1]

**Output:** 8

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

# 278. First Bad Version(easy)

Easy

739453FavoriteShare

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.

**Example:**

Given n = 5, and version = 4 is the first bad version.

call isBadVersion(3) -> false

call isBadVersion(5) -> true

call isBadVersion(4) -> true

Then 4 is the first bad version.

# 283. Move Zeroes(easy)

Easy

231983FavoriteShare

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.

**Example:**

**Input:** [0,1,0,3,12]

**Output:** [1,3,12,0,0]

**Note**:

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

# 290. Word Pattern(easy)

Easy

73093FavoriteShare

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.

**Example 1:**

**Input:** pattern = "abba", str = "dog cat cat dog"

**Output:** true

**Example 2:**

**Input:**pattern = "abba", str = "dog cat cat fish"

**Output:** false

**Example 3:**

**Input:** pattern = "aaaa", str = "dog cat cat dog"

**Output:** false

**Example 4:**

**Input:** pattern = "abba", str = "dog dog dog dog"

**Output:** false

**Notes:**  
You may assume pattern contains only lowercase letters, and strcontains lowercase letters that may be separated by a single space.

# 292. Nim Game(easy)

Easy

4071217FavoriteShare

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.

**Example:**

**Input:** 4

**Output:** false

**Explanation:** 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.

# 299. Bulls and Cows(easy)

Easy

386404FavoriteShare

You are playing the following [Bulls and Cows](https://en.wikipedia.org/wiki/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.

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.

Please note that both secret number and friend's guess may contain duplicate digits.

**Example 1:**

**Input:** secret = "1807", guess = "7810"

**Output:** "1A3B"

**Explanation:** 1 bull and 3 cows. The bull is 8, the cows are 0, 1 and 7.

**Example 2:**

**Input:** secret = "1123", guess = "0111"

**Output:** "1A1B"

**Explanation:** The 1st 1 in friend's guess is a bull, the 2nd or 3rd 1 is a cow.

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

# 303. Range Sum Query – Immutable(easy)

Easy

537837FavoriteShare

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:**

1. You may assume that the array does not change.
2. There are many calls to *sumRange* function.

# 326. Power of Three(easy)

Easy

3241153FavoriteShare

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

**Example 1:**

**Input:** 27

**Output:** true

**Example 2:**

**Input:** 0

**Output:** false

**Example 3:**

**Input:** 9

**Output:** true

**Example 4:**

**Input:** 45

**Output:** false

**Follow up:**  
Could you do it without using any loop / recursion?

# 342. Power of Four(easy)

Easy

341158FavoriteShare

Given an integer (signed 32 bits), write a function to check whether it is a power of 4.

**Example 1:**

**Input:** 16

**Output:** true

**Example 2:**

**Input:** 5

**Output:** false

**Follow up**: Could you solve it without loops/recursion?

# 344. Reverse String(easy)

Easy

863559FavoriteShare

Write a function that reverses a string. The input string is given as an array of characters char[].

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

You may assume all the characters consist of [printable ascii characters](https://en.wikipedia.org/wiki/ASCII#Printable_characters).

**Example 1:**

**Input:** ["h","e","l","l","o"]

**Output:** ["o","l","l","e","h"]

**Example 2:**

**Input:** ["H","a","n","n","a","h"]

**Output:** ["h","a","n","n","a","H"]

# 345. Reverse Vowels of a String(easy)

Easy

433818FavoriteShare

Write a function that takes a string as input and reverse only the vowels of a string.

**Example 1:**

**Input:** "hello"

**Output:** "holle"

**Example 2:**

**Input:** "leetcode"

**Output:** "leotcede"

**Note:**  
The vowels does not include the letter "y".

# 349. Intersection of Two Arrays(easy)

Easy

470881FavoriteShare

Given two arrays, write a function to compute their intersection.

**Example 1:**

**Input:** nums1 = [1,2,2,1], nums2 = [2,2]

**Output:** [2]

**Example 2:**

**Input:** nums1 = [4,9,5], nums2 = [9,4,9,8,4]

**Output:** [9,4]

**Note:**

* Each element in the result must be unique.
* The result can be in any order.

# 350. Intersection of Two Arrays II(easy)

Easy

809287FavoriteShare

Given two arrays, write a function to compute their intersection.

**Example 1:**

**Input:** nums1 = [1,2,2,1], nums2 = [2,2]

**Output:** [2,2]

**Example 2:**

**Input:** nums1 = [4,9,5], nums2 = [9,4,9,8,4]

**Output:** [4,9]

**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?

# 367. Valid Perfect Square(easy)

Easy

516114FavoriteShare

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

**Output:** true

**Example 2:**

**Input:** 14

**Output:** false

# 371. Sum of Two Integers(easy)

Easy

8511578FavoriteShare

Calculate the sum of two integers *a* and *b*, but you are **not allowed** to use the operator + and -.

**Example 1:**

**Input:** a = 1, b = 2

**Output:** 3

**Example 2:**

**Input:** a = -2, b = 3

**Output:** 1

# 374. Guess Number Higher or Lower(easy)

Easy

2621299FavoriteShare

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 :**

**Input:** n = 10, pick = 6

**Output:** 6

# 383. Ransom Note(easy)

Easy

359136FavoriteShare

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

# 387. First Unique Character in a String(easy)

Easy

118886FavoriteShare

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.

# 389. Find the Difference(easy)

Easy

559244FavoriteShare

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.

# 392. Is Subsequence(easy)

Easy

678142FavoriteShare

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](https://leetcode.com/pbrother/) for adding this problem and creating all test cases.

# 401. Binary Watch(easy)

Easy

410692FavoriteShare

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 = 1  
Return: ["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".

# 404. Sum of Left Leaves(easy)

Easy

75084FavoriteShare

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**.

# 405. Convert a Number to Hexadecimal(easy)

Easy

31379FavoriteShare

Given an integer, write an algorithm to convert it to hexadecimal. For negative integer, [two’s complement](https://en.wikipedia.org/wiki/Two%27s_complement) method is used.

**Note:**

1. All letters in hexadecimal (a-f) must be in lowercase.
2. 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.
3. The given number is guaranteed to fit within the range of a 32-bit signed integer.
4. 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"

# 409. Longest Palindrome(easy)

Easy

61164FavoriteShare

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.

# 412. Fizz Buzz(easy)

Easy

597876FavoriteShare

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"

]

# 414. Third Maximum Number(easy)

Easy

430810FavoriteShare

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.

# 415. Add Strings(easy)

Easy

505172FavoriteShare

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

**Note:**

1. The length of both num1 and num2 is < 5100.
2. Both num1 and num2 contains only digits 0-9.
3. Both num1 and num2 does not contain any leading zero.
4. You **must not use any built-in BigInteger library** or **convert the inputs to integer** directly.

# 429. N-ary Tree Level Order Traversal(easy)

Easy

32635FavoriteShare

Given an n-ary tree, return the level order traversal of its nodes' values. (ie, from left to right, level by level).

For example, given a 3-ary tree:



We should return its level order traversal:

[

[1],

[3,2,4],

[5,6]

]

**Note:**

1. The depth of the tree is at most 1000.
2. The total number of nodes is at most 5000.

# 434. Number of Segments in a String(easy)

Easy

154607FavoriteShare

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

# 437. Path Sum III(easy)

Easy

2197126FavoriteShare

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

10

/ \

**5** **-3**

**/** **\** **\**

**3** **2** **11**

/ \ **\**

3 -2 **1**

Return 3. The paths that sum to 8 are:

1. 5 -> 3

2. 5 -> 2 -> 1

3. -3 -> 11

# 441. Arranging Coins(easy)

Easy

222499FavoriteShare

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.

# 443. String Compression(easy)

Easy

4461416FavoriteShare

Given an array of characters, compress it [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm).

The length after compression must always be smaller than or equal to the original array.

Every element of the array should be a **character** (not int) of length 1.

After you are done **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm), return the new length of the array.

**Follow up:**  
Could you solve it using only O(1) extra space?

**Example 1:**

**Input:**

["a","a","b","b","c","c","c"]

**Output:**

Return 6, and the first 6 characters of the input array should be: ["a","2","b","2","c","3"]

**Explanation:**

"aa" is replaced by "a2". "bb" is replaced by "b2". "ccc" is replaced by "c3".

**Example 2:**

**Input:**

["a"]

**Output:**

Return 1, and the first 1 characters of the input array should be: ["a"]

**Explanation:**

Nothing is replaced.

**Example 3:**

**Input:**

["a","b","b","b","b","b","b","b","b","b","b","b","b"]

**Output:**

Return 4, and the first 4 characters of the input array should be: ["a","b","1","2"].

**Explanation:**

Since the character "a" does not repeat, it is not compressed. "bbbbbbbbbbbb" is replaced by "b12".

Notice each digit has it's own entry in the array.

**Note:**

1. All characters have an ASCII value in [35, 126].
2. 1 <= len(chars) <= 1000.

# 447. Number of Boomerangs(easy)

Easy

318500FavoriteShare

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]]**

# 448. Find All Numbers Disappeared in an Array(easy)

Easy

1875175FavoriteShare

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]

# 453. Minimum Moves to Equal Array Elements(easy)

Easy

412636FavoriteShare

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]

# 455. Assign Cookies(easy)

Easy

36369FavoriteShare

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.

# 459. Repeated Substring Pattern(easy)

Easy

906105FavoriteShare

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.)

# 461. Hamming Distance(easy)

Easy

1400133FavoriteShare

The [Hamming distance](https://en.wikipedia.org/wiki/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.

# 463. Island Perimeter(easy)

Easy

122889FavoriteShare

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:**

**Input:**

[[0,1,0,0],

[1,1,1,0],

[0,1,0,0],

[1,1,0,0]]

**Output:** 16

**Explanation:** The perimeter is the 16 yellow stripes in the image below:



# 475. Heaters(easy)

Easy

535577FavoriteShare

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:**

1. Numbers of houses and heaters you are given are non-negative and will not exceed 25000.
2. Positions of houses and heaters you are given are non-negative and will not exceed 10^9.
3. As long as a house is in the heaters' warm radius range, it can be warmed.
4. 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.

# 476. Number Complement(easy)

Easy

58876FavoriteShare

Given a positive integer, output its complement number. The complement strategy is to flip the bits of its binary representation.

**Note:**

1. The given integer is guaranteed to fit within the range of a 32-bit signed integer.
2. 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.

# 482. License Key Formatting(easy)

Easy

334571FavoriteShare

You are given a license key represented as a string S which consists only alphanumeric character and dashes. The string is separated into N+1 groups by N dashes.

Given a number K, we would want to reformat the strings such that each group contains *exactly* K characters, except for the first group which could be shorter than K, but still must contain at least one character. Furthermore, there must be a dash inserted between two groups and all lowercase letters should be converted to uppercase.

Given a non-empty string S and a number K, format the string according to the rules described above.

**Example 1:**

**Input:** S = "5F3Z-2e-9-w", K = 4

**Output:** "5F3Z-2E9W"

**Explanation:** The string S has been split into two parts, each part has 4 characters.

Note that the two extra dashes are not needed and can be removed.

**Example 2:**

**Input:** S = "2-5g-3-J", K = 2

**Output:** "2-5G-3J"

**Explanation:** The string S has been split into three parts, each part has 2 characters except the first part as it could be shorter as mentioned above.

**Note:**

1. The length of string S will not exceed 12,000, and K is a positive integer.
2. String S consists only of alphanumerical characters (a-z and/or A-Z and/or 0-9) and dashes(-).
3. String S is non-empty.

# 485. Max Consecutive Ones(easy)

Easy

413328FavoriteShare

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

# 492. Construct the Rectangle(easy)

Easy

147224FavoriteShare

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:**

1. The given area won't exceed 10,000,000 and is a positive integer
2. The web page's width and length you designed must be positive integers.

# 496. Next Greater Element I(easy)

Easy

9321490FavoriteShare

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:**

1. All elements in nums1 and nums2 are unique.
2. The length of both nums1 and nums2 would not exceed 1000.

# 500. Keyboard Row(easy)

Easy

433544FavoriteShare

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:**

**Input:** ["Hello", "Alaska", "Dad", "Peace"]

**Output:** ["Alaska", "Dad"]

**Note:**

1. You may use one character in the keyboard more than once.
2. You may assume the input string will only contain letters of alphabet.

# 501. Find Mode in Binary Search Tree(easy)

Easy

629251FavoriteShare

Given a binary search tree (BST) with duplicates, find all the [mode(s)](https://en.wikipedia.org/wiki/Mode_(statistics)) (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],

1

\

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).

# 504. Base 7(easy)

Easy

168127FavoriteShare

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].

# 506. Relative Ranks(easy)

Easy

202419FavoriteShare

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:**

1. N is a positive integer and won't exceed 10,000.
2. All the scores of athletes are guaranteed to be unique.

# 507. Perfect Number(easy)

Easy

180491FavoriteShare

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)

# 509. Fibonacci Number(easy)

Easy

281161FavoriteShare

The **Fibonacci numbers**, commonly denoted F(n) form a sequence, called the **Fibonacci sequence**, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is,

F(0) = 0,   F(1) = 1

F(N) = F(N - 1) + F(N - 2), for N > 1.

Given N, calculate F(N).

**Example 1:**

**Input:** 2

**Output:** 1

**Explanation:** F(2) = F(1) + F(0) = 1 + 0 = 1.

**Example 2:**

**Input:** 3

**Output:** 2

**Explanation:** F(3) = F(2) + F(1) = 1 + 1 = 2.

**Example 3:**

**Input:** 4

**Output:** 3

**Explanation:** F(4) = F(3) + F(2) = 2 + 1 = 3.

**Note:**

0 ≤ N ≤ 30.

# 520. Detect Capital(easy)

Easy

344219FavoriteShare

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:

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

# 521. Longest Uncommon Subsequence I(easy)

Easy

2243602FavoriteShare

Given a group of two strings, you need to find the longest uncommon subsequence of this group of two strings. 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 two 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"

**Output:** 3

**Explanation:** The longest uncommon subsequence is "aba" (or "cdc"),   
because "aba" is a subsequence of "aba",   
but not a subsequence of any other strings in the group of two strings.

**Note:**

1. Both strings' lengths will not exceed 100.
2. Only letters from a ~ z will appear in input strings.

# 530. Minimum Absolute Difference in BST(easy)

Easy

60645FavoriteShare

Given a binary search tree with non-negative values, find the minimum [absolute difference](https://en.wikipedia.org/wiki/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.

# 532. K-diff Pairs in an Array(easy)

Easy

401930FavoriteShare

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](https://en.wikipedia.org/wiki/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:**

1. The pairs (i, j) and (j, i) count as the same pair.
2. The length of the array won't exceed 10,000.
3. All the integers in the given input belong to the range: [-1e7, 1e7].

# 538. Convert BST to Greater Tree(easy)

Easy

146693FavoriteShare

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:

5

/ \

2 13

**Output:** The root of a Greater Tree like this:

18

/ \

20 13

# 541. Reverse String II(easy)

Easy

282872FavoriteShare

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:**

1. The string consists of lower English letters only.
2. Length of the given string and k will in the range [1, 10000]

# 543. Diameter of Binary Tree(easy)

Easy

1716107FavoriteShare

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

1

/ \

2 3

/ \

4 5

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.

# 551. Student Attendance Record I(easy)

Easy

175655FavoriteShare

You are given a string representing an attendance record for a student. The record only contains the following three characters:

1. **'A'** : Absent.
2. **'L'** : Late.
3. **'P'** : Present.

A student could be rewarded if his attendance record doesn't contain **more than one 'A' (absent)** or **more than two continuous 'L' (late)**.

You need to return whether the student could be rewarded according to his attendance record.

**Example 1:**

**Input:** "PPALLP"

**Output:** True

**Example 2:**

**Input:** "PPALLL"

**Output:** False

# 557. Reverse Words in a String III(easy)

Easy

71372FavoriteShare

Given a string, you need to reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

**Example 1:**

**Input:** "Let's take LeetCode contest"

**Output:** "s'teL ekat edoCteeL tsetnoc"

**Note:** In the string, each word is separated by single space and there will not be any extra space in the string.

# 558. Quad Tree Intersection(easy)

Easy

58262FavoriteShare

A quadtree is a tree data in which each internal node has exactly four children: topLeft, topRight, bottomLeft and bottomRight. Quad trees are often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions.

We want to store True/False information in our quad tree. The quad tree is used to represent a N \* N boolean grid. For each node, it will be subdivided into four children nodes **until the values in the region it represents are all the same**. Each node has another two boolean attributes : isLeaf and val. isLeaf is true if and only if the node is a leaf node. The val attribute for a leaf node contains the value of the region it represents.

For example, below are two quad trees A and B:

A:

+-------+-------+ T: true

| | | F: false

| T | T |

| | |

+-------+-------+

| | |

| F | F |

| | |

+-------+-------+

topLeft: T

topRight: T

bottomLeft: F

bottomRight: F

B:

+-------+---+---+

| | F | F |

| T +---+---+

| | T | T |

+-------+---+---+

| | |

| T | F |

| | |

+-------+-------+

topLeft: T

topRight:

topLeft: F

topRight: F

bottomLeft: T

bottomRight: T

bottomLeft: T

bottomRight: F

Your task is to implement a function that will take two quadtrees and return a quadtree that represents the logical OR (or union) of the two trees.

A: B: C (A or B):

+-------+-------+ +-------+---+---+ +-------+-------+

| | | | | F | F | | | |

| T | T | | T +---+---+ | T | T |

| | | | | T | T | | | |

+-------+-------+ +-------+---+---+ +-------+-------+

| | | | | | | | |

| F | F | | T | F | | T | F |

| | | | | | | | |

+-------+-------+ +-------+-------+ +-------+-------+

**Note:**

1. Both A and B represent grids of size N \* N.
2. N is guaranteed to be a power of 2.
3. If you want to know more about the quad tree, you can refer to its [wiki](https://en.wikipedia.org/wiki/Quadtree).
4. The logic OR operation is defined as this: "A or B" is true if A is true, or if B is true, or if both A and B are true.

# 559. Maximum Depth of N-ary Tree(easy)

Easy

46026FavoriteShare

Given a n-ary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

For example, given a 3-ary tree:



We should return its max depth, which is 3.

**Note:**

1. The depth of the tree is at most 1000.
2. The total number of nodes is at most 5000.

# 561. Array Partition I(easy)

Easy

6231933FavoriteShare

Given an array of **2n** integers, your task is to group these integers into **n** pairs of integer, say (a1, b1), (a2, b2), ..., (an, bn) which makes sum of min(ai, bi) for all i from 1 to n as large as possible.

**Example 1:**

**Input:** [1,4,3,2]

**Output:** 4

**Explanation:** n is 2, and the maximum sum of pairs is 4 = min(1, 2) + min(3, 4).

**Note:**

1. **n** is a positive integer, which is in the range of [1, 10000].
2. All the integers in the array will be in the range of [-10000, 10000].

# 563. Binary Tree Tilt(easy)

Easy

363847FavoriteShare

Given a binary tree, return the tilt of the **whole tree**.

The tilt of a **tree node** is defined as the **absolute difference** between the sum of all left subtree node values and the sum of all right subtree node values. Null node has tilt 0.

The tilt of the **whole tree** is defined as the sum of all nodes' tilt.

**Example:**

**Input:**

1

/ \

2 3

**Output:** 1

**Explanation:**

Tilt of node 2 : 0

Tilt of node 3 : 0

Tilt of node 1 : |2-3| = 1

Tilt of binary tree : 0 + 0 + 1 = 1

**Note:**

1. The sum of node values in any subtree won't exceed the range of 32-bit integer.
2. All the tilt values won't exceed the range of 32-bit integer.

# 566. Reshape the Matrix(easy)

Easy

62691FavoriteShare

In MATLAB, there is a very useful function called 'reshape', which can reshape a matrix into a new one with different size but keep its original data.

You're given a matrix represented by a two-dimensional array, and two **positive** integers **r** and **c** representing the **row** number and **column** number of the wanted reshaped matrix, respectively.

The reshaped matrix need to be filled with all the elements of the original matrix in the same **row-traversing** order as they were.

If the 'reshape' operation with given parameters is possible and legal, output the new reshaped matrix; Otherwise, output the original matrix.

**Example 1:**

**Input:**

nums =

[[1,2],

[3,4]]

r = 1, c = 4

**Output:**

[[1,2,3,4]]

**Explanation:**  
The **row-traversing** of nums is [1,2,3,4]. The new reshaped matrix is a 1 \* 4 matrix, fill it row by row by using the previous list.

**Example 2:**

**Input:**

nums =

[[1,2],

[3,4]]

r = 2, c = 4

**Output:**

[[1,2],

[3,4]]

**Explanation:**  
There is no way to reshape a 2 \* 2 matrix to a 2 \* 4 matrix. So output the original matrix.

**Note:**

1. The height and width of the given matrix is in range [1, 100].
2. The given r and c are all positive.

# 572. Subtree of Another Tree(easy)

Easy

141254FavoriteShare

Given two non-empty binary trees **s** and **t**, check whether tree **t** has exactly the same structure and node values with a subtree of **s**. A subtree of **s** is a tree consists of a node in **s** and all of this node's descendants. The tree **s** could also be considered as a subtree of itself.

**Example 1:**  
Given tree s:

3

/ \

4 5

/ \

1 2

Given tree t:

4

/ \

1 2

Return **true**, because t has the same structure and node values with a subtree of s.

**Example 2:**  
Given tree s:

3

/ \

4 5

/ \

1 2

/

0

Given tree t:

4

/ \

1 2

Return **false**.

# 575. Distribute Candies(easy)

Easy

330734FavoriteShare

Given an integer array with **even** length, where different numbers in this array represent different **kinds** of candies. Each number means one candy of the corresponding kind. You need to distribute these candies **equally** in number to brother and sister. Return the maximum number of **kinds** of candies the sister could gain.

**Example 1:**

**Input:** candies = [1,1,2,2,3,3]

**Output:** 3

**Explanation:**

There are three different kinds of candies (1, 2 and 3), and two candies for each kind.

Optimal distribution: The sister has candies [1,2,3] and the brother has candies [1,2,3], too.

The sister has three different kinds of candies.

**Example 2:**

**Input:** candies = [1,1,2,3]

**Output:** 2

**Explanation:** For example, the sister has candies [2,3] and the brother has candies [1,1].

The sister has two different kinds of candies, the brother has only one kind of candies.

**Note:**

1. The length of the given array is in range [2, 10,000], and will be even.
2. The number in given array is in range [-100,000, 100,000].

# 581. Shortest Unsorted Continuous Subarray(easy)

Easy

172278FavoriteShare

Given an integer array, you need to find one **continuous subarray** that if you only sort this subarray in ascending order, then the whole array will be sorted in ascending order, too.

You need to find the **shortest** such subarray and output its length.

**Example 1:**

**Input:** [2, 6, 4, 8, 10, 9, 15]

**Output:** 5

**Explanation:** You need to sort [6, 4, 8, 10, 9] in ascending order to make the whole array sorted in ascending order.

**Note:**

1. Then length of the input array is in range [1, 10,000].
2. The input array may contain duplicates, so ascending order here means **<=**.

# 589. N-ary Tree Preorder Traversal(easy)

Easy

30143FavoriteShare

Given an n-ary tree, return the *preorder* traversal of its nodes' values.

For example, given a 3-ary tree:



Return its preorder traversal as: [1,3,5,6,2,4].

**Note:**

Recursive solution is trivial, could you do it iteratively?

# 590. N-ary Tree Postorder Traversal(easy)

Easy

37546FavoriteShare

Given an n-ary tree, return the *postorder* traversal of its nodes' values.

For example, given a 3-ary tree:



Return its postorder traversal as: [5,6,3,2,4,1].

**Note:**

Recursive solution is trivial, could you do it iteratively?

# 594. Longest Harmonious Subsequence(easy)

Easy

46665FavoriteShare

We define a harmounious array as an array where the difference between its maximum value and its minimum value is **exactly** 1.

Now, given an integer array, you need to find the length of its longest harmonious subsequence among all its possible [subsequences](https://en.wikipedia.org/wiki/Subsequence).

**Example 1:**

**Input:** [1,3,2,2,5,2,3,7]

**Output:** 5

**Explanation:** The longest harmonious subsequence is [3,2,2,2,3].

**Note:** The length of the input array will not exceed 20,000.

# 595. Big Countries(easy)

Easy

398566FavoriteShare

SQL Schema

There is a table World

+-----------------+------------+------------+--------------+---------------+

| name | continent | area | population | gdp |

+-----------------+------------+------------+--------------+---------------+

| Afghanistan | Asia | 652230 | 25500100 | 20343000 |

| Albania | Europe | 28748 | 2831741 | 12960000 |

| Algeria | Africa | 2381741 | 37100000 | 188681000 |

| Andorra | Europe | 468 | 78115 | 3712000 |

| Angola | Africa | 1246700 | 20609294 | 100990000 |

+-----------------+------------+------------+--------------+---------------+

A country is big if it has an area of bigger than 3 million square km or a population of more than 25 million.

Write a SQL solution to output big countries' name, population and area.

For example, according to the above table, we should output:

+--------------+-------------+--------------+

| name | population | area |

+--------------+-------------+--------------+

| Afghanistan | 25500100 | 652230 |

| Algeria | 37100000 | 2381741 |

+--------------+-------------+--------------+

# 596. Classes More Than 5 Students(easy)

Easy

169490FavoriteShare

SQL Schema

There is a table courses with columns: **student** and **class**

Please list out all classes which have more than or equal to 5 students.

For example, the table:

+---------+------------+

| student | class |

+---------+------------+

| A | Math |

| B | English |

| C | Math |

| D | Biology |

| E | Math |

| F | Computer |

| G | Math |

| H | Math |

| I | Math |

+---------+------------+

Should output:

+---------+

| class |

+---------+

| Math |

+---------+

**Note:**  
The students should not be counted duplicate in each course.

# 598. Range Addition II(easy)

Easy

213493FavoriteShare

Given an m \* n matrix **M** initialized with all **0**'s and several update operations.

Operations are represented by a 2D array, and each operation is represented by an array with two **positive** integers **a** and **b**, which means **M[i][j]** should be **added by one** for all **0 <= i < a** and **0 <= j < b**.

You need to count and return the number of maximum integers in the matrix after performing all the operations.

**Example 1:**

**Input:**

m = 3, n = 3

operations = [[2,2],[3,3]]

**Output:** 4

**Explanation:**

Initially, M =

[[0, 0, 0],

[0, 0, 0],

[0, 0, 0]]

After performing [2,2], M =

[[1, 1, 0],

[1, 1, 0],

[0, 0, 0]]

After performing [3,3], M =

[[2, 2, 1],

[2, 2, 1],

[1, 1, 1]]

So the maximum integer in M is 2, and there are four of it in M. So return 4.

**Note:**

1. The range of m and n is [1,40000].
2. The range of a is [1,m], and the range of b is [1,n].
3. The range of operations size won't exceed 10,000.

# 599. Minimum Index Sum of Two Lists(easy)

Easy

413159FavoriteShare

Suppose Andy and Doris want to choose a restaurant for dinner, and they both have a list of favorite restaurants represented by strings.

You need to help them find out their **common interest** with the **least list index sum**. If there is a choice tie between answers, output all of them with no order requirement. You could assume there always exists an answer.

**Example 1:**

**Input:**

["Shogun", "Tapioca Express", "Burger King", "KFC"]

["Piatti", "The Grill at Torrey Pines", "Hungry Hunter Steakhouse", "Shogun"]

**Output:** ["Shogun"]

**Explanation:** The only restaurant they both like is "Shogun".

**Example 2:**

**Input:**

["Shogun", "Tapioca Express", "Burger King", "KFC"]

["KFC", "Shogun", "Burger King"]

**Output:** ["Shogun"]

**Explanation:** The restaurant they both like and have the least index sum is "Shogun" with index sum 1 (0+1).

**Note:**

1. The length of both lists will be in the range of [1, 1000].
2. The length of strings in both lists will be in the range of [1, 30].
3. The index is starting from 0 to the list length minus 1.
4. No duplicates in both lists.

# 605. Can Place Flowers(easy)

Easy

559290FavoriteShare

Suppose you have a long flowerbed in which some of the plots are planted and some are not. However, flowers cannot be planted in adjacent plots - they would compete for water and both would die.

Given a flowerbed (represented as an array containing 0 and 1, where 0 means empty and 1 means not empty), and a number **n**, return if **n** new flowers can be planted in it without violating the no-adjacent-flowers rule.

**Example 1:**

**Input:** flowerbed = [1,0,0,0,1], n = 1

**Output:** True

**Example 2:**

**Input:** flowerbed = [1,0,0,0,1], n = 2

**Output:** False

**Note:**

1. The input array won't violate no-adjacent-flowers rule.
2. The input array size is in the range of [1, 20000].
3. **n** is a non-negative integer which won't exceed the input array size.

# 606. Construct String from Binary Tree(easy)

Easy

592818FavoriteShare

You need to construct a string consists of parenthesis and integers from a binary tree with the preorder traversing way.

The null node needs to be represented by empty parenthesis pair "()". And you need to omit all the empty parenthesis pairs that don't affect the one-to-one mapping relationship between the string and the original binary tree.

**Example 1:**

**Input:** Binary tree: [1,2,3,4]

1

/ \

2 3

/

4

**Output:** "1(2(4))(3)"

**Explanation:** Originallay it needs to be "1(2(4)())(3()())",   
but you need to omit all the unnecessary empty parenthesis pairs.   
And it will be "1(2(4))(3)".

**Example 2:**

**Input:** Binary tree: [1,2,3,null,4]

1

/ \

2 3

\

4

**Output:** "1(2()(4))(3)"

**Explanation:** Almost the same as the first example,   
except we can't omit the first parenthesis pair to break the one-to-one mapping relationship between the input and the output.

# 617. Merge Two Binary Trees(easy)

Easy

2100143FavoriteShare

Given two binary trees and imagine that when you put one of them to cover the other, some nodes of the two trees are overlapped while the others are not.

You need to merge them into a new binary tree. The merge rule is that if two nodes overlap, then sum node values up as the new value of the merged node. Otherwise, the NOT null node will be used as the node of new tree.

**Example 1:**

**Input:**

Tree 1 Tree 2

1 2

/ \ / \

3 2 1 3

/ \ \

5 4 7

**Output:**

Merged tree:

3

/ \

4 5

/ \ \

5 4 7

**Note:** The merging process must start from the root nodes of both trees.

# 620. Not Boring Movies(easy)

Easy

197230FavoriteShare

SQL Schema

X city opened a new cinema, many people would like to go to this cinema. The cinema also gives out a poster indicating the movies’ ratings and descriptions.

Please write a SQL query to output movies with an odd numbered ID and a description that is not 'boring'. Order the result by rating.

For example, table cinema:

+---------+-----------+--------------+-----------+

| id | movie | description | rating |

+---------+-----------+--------------+-----------+

| 1 | War | great 3D | 8.9 |

| 2 | Science | fiction | 8.5 |

| 3 | irish | boring | 6.2 |

| 4 | Ice song | Fantacy | 8.6 |

| 5 | House card| Interesting| 9.1 |

+---------+-----------+--------------+-----------+

For the example above, the output should be:

+---------+-----------+--------------+-----------+

| id | movie | description | rating |

+---------+-----------+--------------+-----------+

| 5 | House card| Interesting| 9.1 |

| 1 | War | great 3D | 8.9 |

+---------+-----------+--------------+-----------+

# 627. Swap Salary(easy)

Easy

348249FavoriteShare

SQL Schema

Given a table salary, such as the one below, that has m=male and f=female values. Swap all f and m values (i.e., change all f values to m and vice versa) with a **single update statement** and no intermediate temp table.

Note that you must write a single update statement, **DO NOT** write any select statement for this problem.

**Example:**

| id | name | sex | salary |

|----|------|-----|--------|

| 1 | A | m | 2500 |

| 2 | B | f | 1500 |

| 3 | C | m | 5500 |

| 4 | D | f | 500 |

After running your **update** statement, the above salary table should have the following rows:

| id | name | sex | salary |

|----|------|-----|--------|

| 1 | A | f | 2500 |

| 2 | B | m | 1500 |

| 3 | C | f | 5500 |

| 4 | D | m | 500 |

# 628. Maximum Product of Three Numbers(easy)

Easy

792291FavoriteShare

Given an integer array, find three numbers whose product is maximum and output the maximum product.

**Example 1:**

**Input:** [1,2,3]

**Output:** 6

**Example 2:**

**Input:** [1,2,3,4]

**Output:** 24

**Note:**

1. The length of the given array will be in range [3,104] and all elements are in the range [-1000, 1000].
2. Multiplication of any three numbers in the input won't exceed the range of 32-bit signed integer.

# 633. Sum of Square Numbers(easy)

Easy

381255FavoriteShare

Given a non-negative integer c, your task is to decide whether there're two integers a and b such that a2 + b2 = c.

**Example 1:**

**Input:** 5

**Output:** True

**Explanation:** 1 \* 1 + 2 \* 2 = 5

**Example 2:**

**Input:** 3

**Output:** False

# 637. Average of Levels in Binary Tree(easy)

Easy

900130FavoriteShare

Given a non-empty binary tree, return the average value of the nodes on each level in the form of an array.

**Example 1:**

**Input:**

3

/ \

9 20

/ \

15 7

**Output:** [3, 14.5, 11]

**Explanation:**

The average value of nodes on level 0 is 3, on level 1 is 14.5, and on level 2 is 11. Hence return [3, 14.5, 11].

**Note:**

1. The range of node's value is in the range of 32-bit signed integer.

# 643. Maximum Average Subarray I(easy)

Easy

51689FavoriteShare

Given an array consisting of n integers, find the contiguous subarray of given length k that has the maximum average value. And you need to output the maximum average value.

**Example 1:**

**Input:** [1,12,-5,-6,50,3], k = 4

**Output:** 12.75

**Explanation:** Maximum average is (12-5-6+50)/4 = 51/4 = 12.75

**Note:**

1. 1 <= k <= n <= 30,000.
2. Elements of the given array will be in the range [-10,000, 10,000].

# 645. Set Mismatch(easy)

Easy

472258FavoriteShare

The set S originally contains numbers from 1 to n. But unfortunately, due to the data error, one of the numbers in the set got duplicated to **another** number in the set, which results in repetition of one number and loss of another number.

Given an array nums representing the data status of this set after the error. Your task is to firstly find the number occurs twice and then find the number that is missing. Return them in the form of an array.

**Example 1:**

**Input:** nums = [1,2,2,4]

**Output:** [2,3]

**Note:**

1. The given array size will in the range [2, 10000].
2. The given array's numbers won't have any order.

Accepted

55,294

Submissions

134,198

# 653. Two Sum IV - Input is a BST(easy)

Easy

1015117FavoriteShare

Given a Binary Search Tree and a target number, return true if there exist two elements in the BST such that their sum is equal to the given target.

**Example 1:**

**Input:**

5

/ \

3 6

/ \ \

2 4 7

Target = 9

**Output:** True

**Example 2:**

**Input:**

5

/ \

3 6

/ \ \

2 4 7

Target = 28

**Output:** False

# 657. Robot Return to Origin(easy)

Easy

766587FavoriteShare

There is a robot starting at position (0, 0), the origin, on a 2D plane. Given a sequence of its moves, judge if this robot **ends up at (0, 0)** after it completes its moves.

The move sequence is represented by a string, and the character moves[i] represents its ith move. Valid moves are R (right), L (left), U (up), and D (down). If the robot returns to the origin after it finishes all of its moves, return true. Otherwise, return false.

**Note**: The way that the robot is "facing" is irrelevant. "R" will always make the robot move to the right once, "L" will always make it move left, etc. Also, assume that the magnitude of the robot's movement is the same for each move.

**Example 1:**

**Input:** "UD"

**Output:** true

**Explanation**: The robot moves up once, and then down once. All moves have the same magnitude, so it ended up at the origin where it started. Therefore, we return true.

**Example 2:**

**Input:** "LL"

**Output:** false

**Explanation**: The robot moves left twice. It ends up two "moves" to the left of the origin. We return false because it is not at the origin at the end of its moves.

# 661. Image Smoother(easy)

Easy

195884FavoriteShare

Given a 2D integer matrix M representing the gray scale of an image, you need to design a smoother to make the gray scale of each cell becomes the average gray scale (rounding down) of all the 8 surrounding cells and itself. If a cell has less than 8 surrounding cells, then use as many as you can.

**Example 1:**

**Input:**

[[1,1,1],

[1,0,1],

[1,1,1]]

**Output:**

[[0, 0, 0],

[0, 0, 0],

[0, 0, 0]]

**Explanation:**

For the point (0,0), (0,2), (2,0), (2,2): floor(3/4) = floor(0.75) = 0

For the point (0,1), (1,0), (1,2), (2,1): floor(5/6) = floor(0.83333333) = 0

For the point (1,1): floor(8/9) = floor(0.88888889) = 0

**Note:**

1. The value in the given matrix is in the range of [0, 255].
2. The length and width of the given matrix are in the range of [1, 150].

# 665. Non-decreasing Array(easy)

Easy

1258291FavoriteShare

Given an array with n integers, your task is to check if it could become non-decreasing by modifying **at most** 1 element.

We define an array is non-decreasing if array[i] <= array[i + 1] holds for every i (1 <= i < n).

**Example 1:**

**Input:** [4,2,3]

**Output:** True

**Explanation:** You could modify the first 4 to 1 to get a non-decreasing array.

**Example 2:**

**Input:** [4,2,1]

**Output:** False

**Explanation:** You can't get a non-decreasing array by modify at most one element.

**Note:** The n belongs to [1, 10,000].

# 669. Trim a Binary Search Tree(easy)

Easy

1295140FavoriteShare

Given a binary search tree and the lowest and highest boundaries as L and R, trim the tree so that all its elements lies in [L, R] (R >= L). You might need to change the root of the tree, so the result should return the new root of the trimmed binary search tree.

**Example 1:**

**Input:**

1

/ \

0 2

L = 1

R = 2

**Output:**

1

\

2

**Example 2:**

**Input:**

3

/ \

0 4

\

2

/

1

L = 1

R = 3

**Output:**

3

/

2

/

1

# 671. Second Minimum Node In a Binary Tree(easy)

Easy

438658FavoriteShare

Given a non-empty special binary tree consisting of nodes with the non-negative value, where each node in this tree has exactly two or zero sub-node. If the node has two sub-nodes, then this node's value is the smaller value among its two sub-nodes. More formally, the property root.val = min(root.left.val, root.right.val) always holds.

Given such a binary tree, you need to output the **second minimum** value in the set made of all the nodes' value in the whole tree.

If no such second minimum value exists, output -1 instead.

**Example 1:**

**Input:**

2

/ \

2 5

/ \

5 7

**Output:** 5

**Explanation:** The smallest value is 2, the second smallest value is 5.

**Example 2:**

**Input:**

2

/ \

2 2

**Output:** -1

**Explanation:** The smallest value is 2, but there isn't any second smallest value.

# 674. Longest Continuous Increasing Subsequence(easy)

Easy

508101FavoriteShare

Given an unsorted array of integers, find the length of longest continuous increasing subsequence (subarray).

**Example 1:**

**Input:** [1,3,5,4,7]

**Output:** 3

**Explanation:** The longest continuous increasing subsequence is [1,3,5], its length is 3.

Even though [1,3,5,7] is also an increasing subsequence, it's not a continuous one where 5 and 7 are separated by 4.

**Example 2:**

**Input:** [2,2,2,2,2]

**Output:** 1

**Explanation:** The longest continuous increasing subsequence is [2], its length is 1.

**Note:** Length of the array will not exceed 10,000.

# 680. Valid Palindrome II(easy)

Easy

93367FavoriteShare

Given a non-empty string s, you may delete **at most** one character. Judge whether you can make it a palindrome.

**Example 1:**

**Input:** "aba"

**Output:** True

**Example 2:**

**Input:** "abca"

**Output:** True

**Explanation:** You could delete the character 'c'.

**Note:**

1. The string will only contain lowercase characters a-z. The maximum length of the string is 50000.

# 682. Baseball Game(easy)

Easy

347856FavoriteShare

You're now a baseball game point recorder.

Given a list of strings, each string can be one of the 4 following types:

1. Integer (one round's score): Directly represents the number of points you get in this round.
2. "+" (one round's score): Represents that the points you get in this round are the sum of the last two valid round's points.
3. "D" (one round's score): Represents that the points you get in this round are the doubled data of the last valid round's points.
4. "C" (an operation, which isn't a round's score): Represents the last valid round's points you get were invalid and should be removed.

Each round's operation is permanent and could have an impact on the round before and the round after.

You need to return the sum of the points you could get in all the rounds.

**Example 1:**

**Input:** ["5","2","C","D","+"]

**Output:** 30

**Explanation:**

Round 1: You could get 5 points. The sum is: 5.

Round 2: You could get 2 points. The sum is: 7.

Operation 1: The round 2's data was invalid. The sum is: 5.

Round 3: You could get 10 points (the round 2's data has been removed). The sum is: 15.

Round 4: You could get 5 + 10 = 15 points. The sum is: 30.

**Example 2:**

**Input:** ["5","-2","4","C","D","9","+","+"]

**Output:** 27

**Explanation:**

Round 1: You could get 5 points. The sum is: 5.

Round 2: You could get -2 points. The sum is: 3.

Round 3: You could get 4 points. The sum is: 7.

Operation 1: The round 3's data is invalid. The sum is: 3.

Round 4: You could get -4 points (the round 3's data has been removed). The sum is: -1.

Round 5: You could get 9 points. The sum is: 8.

Round 6: You could get -4 + 9 = 5 points. The sum is 13.

Round 7: You could get 9 + 5 = 14 points. The sum is 27.

**Note:**

 The size of the input list will be between 1 and 1000.

 Every integer represented in the list will be between -30000 and 30000.

# 686. Repeated String Match(easy)

Easy

569573FavoriteShare

Given two strings A and B, find the minimum number of times A has to be repeated such that B is a substring of it. If no such solution, return -1.

For example, with A = "abcd" and B = "cdabcdab".

Return 3, because by repeating A three times (“abcdabcdabcd”), B is a substring of it; and B is not a substring of A repeated two times ("abcdabcd").

**Note:**  
The length of A and B will be between 1 and 10000.

# 687. Longest Univalue Path(easy)

Easy

1152301FavoriteShare

Given a binary tree, find the length of the longest path where each node in the path has the same value. This path may or may not pass through the root.

The length of path between two nodes is represented by the number of edges between them.

**Example 1:**

**Input:**

5

/ \

4 5

/ \ \

1 1 5

**Output:** 2

**Example 2:**

**Input:**

1

/ \

4 5

/ \ \

4 4 5

**Output:** 2

**Note:** The given binary tree has not more than 10000 nodes. The height of the tree is not more than 1000.

# 690. Employee Importance(easy)

Easy

451481FavoriteShare

You are given a data structure of employee information, which includes the employee's **unique id**, his **importance value** and his **direct** subordinates' id.

For example, employee 1 is the leader of employee 2, and employee 2 is the leader of employee 3. They have importance value 15, 10 and 5, respectively. Then employee 1 has a data structure like [1, 15, [2]], and employee 2 has [2, 10, [3]], and employee 3 has [3, 5, []]. Note that although employee 3 is also a subordinate of employee 1, the relationship is **not direct**.

Now given the employee information of a company, and an employee id, you need to return the total importance value of this employee and all his subordinates.

**Example 1:**

**Input:** [[1, 5, [2, 3]], [2, 3, []], [3, 3, []]], 1

**Output:** 11

**Explanation:**

Employee 1 has importance value 5, and he has two direct subordinates: employee 2 and employee 3. They both have importance value 3. So the total importance value of employee 1 is 5 + 3 + 3 = 11.

**Note:**

1. One employee has at most one **direct** leader and may have several subordinates.
2. The maximum number of employees won't exceed 2000.

# 693. Binary Number with Alternating Bits(easy)

Easy

33174FavoriteShare

Given a positive integer, check whether it has alternating bits: namely, if two adjacent bits will always have different values.

**Example 1:**

**Input:** 5

**Output:** True

**Explanation:**

The binary representation of 5 is: 101

**Example 2:**

**Input:** 7

**Output:** False

**Explanation:**

The binary representation of 7 is: 111.

**Example 3:**

**Input:** 11

**Output:** False

**Explanation:**

The binary representation of 11 is: 1011.

**Example 4:**

**Input:** 10

**Output:** True

**Explanation:**

The binary representation of 10 is: 1010.

# 696. Count Binary Substrings(easy)

Easy

716131FavoriteShare

Give a string s, count the number of non-empty (contiguous) substrings that have the same number of 0's and 1's, and all the 0's and all the 1's in these substrings are grouped consecutively.

Substrings that occur multiple times are counted the number of times they occur.

**Example 1:**

**Input:** "00110011"

**Output:** 6

**Explanation:** There are 6 substrings that have equal number of consecutive 1's and 0's: "0011", "01", "1100", "10", "0011", and "01".

Notice that some of these substrings repeat and are counted the number of times they occur.

Also, "00110011" is not a valid substring because **all** the 0's (and 1's) are not grouped together.

**Example 2:**

**Input:** "10101"

**Output:** 4

**Explanation:** There are 4 substrings: "10", "01", "10", "01" that have equal number of consecutive 1's and 0's.

**Note:**

 s.length will be between 1 and 50,000.

 s will only consist of "0" or "1" characters.

# 697. Degree of an Array(easy)

Easy

602538FavoriteShare

Given a non-empty array of non-negative integers nums, the **degree** of this array is defined as the maximum frequency of any one of its elements.

Your task is to find the smallest possible length of a (contiguous) subarray of nums, that has the same degree as nums.

**Example 1:**

**Input:** [1, 2, 2, 3, 1]

**Output:** 2

**Explanation:**

The input array has a degree of 2 because both elements 1 and 2 appear twice.

Of the subarrays that have the same degree:

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

The shortest length is 2. So return 2.

**Example 2:**

**Input:** [1,2,2,3,1,4,2]

**Output:** 6

**Note:**

 nums.length will be between 1 and 50,000.

 nums[i] will be an integer between 0 and 49,999.

# 700. Search in a Binary Search Tree(easy)

Easy

41399FavoriteShare

Given the root node of a binary search tree (BST) and a value. You need to find the node in the BST that the node's value equals the given value. Return the subtree rooted with that node. If such node doesn't exist, you should return NULL.

For example,

Given the tree:

4

/ \

2 7

/ \

1 3

And the value to search: 2

You should return this subtree:

2

/ \

1 3

In the example above, if we want to search the value 5, since there is no node with value 5, we should return NULL.

Note that an empty tree is represented by NULL, therefore you would see the expected output (serialized tree format) as [], not null.

# 703. Kth Largest Element in a Stream(easy)

Easy

408198FavoriteShare

Design a class to find the **k**th largest element in a stream. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Your KthLargest class will have a constructor which accepts an integer k and an integer array nums, which contains initial elements from the stream. For each call to the method KthLargest.add, return the element representing the kth largest element in the stream.

**Example:**

int k = 3;

int[] arr = [4,5,8,2];

KthLargest kthLargest = new KthLargest(3, arr);

kthLargest.add(3);   // returns 4

kthLargest.add(5);   // returns 5

kthLargest.add(10);  // returns 5

kthLargest.add(9);   // returns 8

kthLargest.add(4);   // returns 8

**Note:**  
You may assume that nums' length ≥ k-1 and k ≥ 1.

# 704. Binary Search(easy)

Easy

35135FavoriteShare

Given a **sorted** (in ascending order) integer array nums of n elements and a target value, write a function to search target in nums. If target exists, then return its index, otherwise return -1.

**Example 1:**

**Input:** nums = [-1,0,3,5,9,12], target = 9

**Output:** 4

**Explanation:** 9 exists in nums and its index is 4

**Example 2:**

**Input:** nums = [-1,0,3,5,9,12], target = 2

**Output:** -1

**Explanation:** 2 does not exist in nums so return -1

**Note:**

1. You may assume that all elements in nums are unique.
2. n will be in the range [1, 10000].
3. The value of each element in nums will be in the range [-9999, 9999].

# 705. Design HashSet(easy)

Easy

18449FavoriteShare

Design a HashSet without using any built-in hash table libraries.

To be specific, your design should include these functions:

* add(value): Insert a value into the HashSet.
* contains(value) : Return whether the value exists in the HashSet or not.
* remove(value): Remove a value in the HashSet. If the value does not exist in the HashSet, do nothing.

**Example:**

MyHashSet hashSet = new MyHashSet();

hashSet.add(1);

hashSet.add(2);

hashSet.contains(1);    // returns true

hashSet.contains(3);    // returns false (not found)

hashSet.add(2);

hashSet.contains(2);    // returns true

hashSet.remove(2);

hashSet.contains(2);    // returns false (already removed)

**Note:**

* All values will be in the range of [0, 1000000].
* The number of operations will be in the range of [1, 10000].
* Please do not use the built-in HashSet library.

# 706. Design HashMap(easy)

Easy

46271FavoriteShare

Design a HashMap without using any built-in hash table libraries.

To be specific, your design should include these functions:

* put(key, value) : Insert a (key, value) pair into the HashMap. If the value already exists in the HashMap, update the value.
* get(key): Returns the value to which the specified key is mapped, or -1 if this map contains no mapping for the key.
* remove(key) : Remove the mapping for the value key if this map contains the mapping for the key.

**Example:**

MyHashMap hashMap = new MyHashMap();

hashMap.put(1, 1);

hashMap.put(2, 2);

hashMap.get(1);            // returns 1

hashMap.get(3);            // returns -1 (not found)

hashMap.put(2, 1);          // update the existing value

hashMap.get(2);            // returns 1

hashMap.remove(2);          // remove the mapping for 2

hashMap.get(2);            // returns -1 (not found)

**Note:**

* All keys and values will be in the range of [0, 1000000].
* The number of operations will be in the range of [1, 10000].
* Please do not use the built-in HashMap library.

# 709. To Lower Case(easy)

Easy

3431194FavoriteShare

Implement function ToLowerCase() that has a string parameter str, and returns the same string in lowercase.

**Example 1:**

**Input:** "Hello"

**Output:** "hello"

**Example 2:**

**Input:** "here"

**Output:** "here"

**Example 3:**

**Input:** "LOVELY"

**Output:** "lovely"

# 717. 1-bit and 2-bit Characters(easy)

Easy

290761FavoriteShare

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

Now given a string represented by several bits. Return whether the last character must be a one-bit character or not. The given string will always end with a zero.

**Example 1:**

**Input:**

bits = [1, 0, 0]

**Output:** True

**Explanation:**

The only way to decode it is two-bit character and one-bit character. So the last character is one-bit character.

**Example 2:**

**Input:**

bits = [1, 1, 1, 0]

**Output:** False

**Explanation:**

The only way to decode it is two-bit character and two-bit character. So the last character is NOT one-bit character.

**Note:**

 1 <= len(bits) <= 1000.

 bits[i] is always 0 or 1.

# 720. Longest Word in Dictionary(easy)

Easy

415528FavoriteShare

Given a list of strings words representing an English Dictionary, find the longest word in words that can be built one character at a time by other words in words. If there is more than one possible answer, return the longest word with the smallest lexicographical order.

If there is no answer, return the empty string.

**Example 1:**

**Input:**

words = ["w","wo","wor","worl", "world"]

**Output:** "world"

**Explanation:**

The word "world" can be built one character at a time by "w", "wo", "wor", and "worl".

**Example 2:**

**Input:**

words = ["a", "banana", "app", "appl", "ap", "apply", "apple"]

**Output:** "apple"

**Explanation:**

Both "apply" and "apple" can be built from other words in the dictionary. However, "apple" is lexicographically smaller than "apply".

**Note:**

 All the strings in the input will only contain lowercase letters.

 The length of words will be in the range [1, 1000].

 The length of words[i] will be in the range [1, 30].

# 724. Find Pivot Index(easy)

Easy

751176FavoriteShare

Given an array of integers nums, write a method that returns the "pivot" index of this array.

We define the pivot index as the index where the sum of the numbers to the left of the index is equal to the sum of the numbers to the right of the index.

If no such index exists, we should return -1. If there are multiple pivot indexes, you should return the left-most pivot index.

**Example 1:**

**Input:**

nums = [1, 7, 3, 6, 5, 6]

**Output:** 3

**Explanation:**

The sum of the numbers to the left of index 3 (nums[3] = 6) is equal to the sum of numbers to the right of index 3.

Also, 3 is the first index where this occurs.

**Example 2:**

**Input:**

nums = [1, 2, 3]

**Output:** -1

**Explanation:**

There is no index that satisfies the conditions in the problem statement.

**Note:**

* The length of nums will be in the range [0, 10000].
* Each element nums[i] will be an integer in the range [-1000, 1000].

# 728. Self Dividing Numbers(easy)

Easy

485261FavoriteShare

A *self-dividing number* is a number that is divisible by every digit it contains.

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

Also, a self-dividing number is not allowed to contain the digit zero.

Given a lower and upper number bound, output a list of every possible self dividing number, including the bounds if possible.

**Example 1:**

**Input:**

left = 1, right = 22

**Output:** [1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 22]

**Note:**

 The boundaries of each input argument are 1 <= left <= right <= 10000.

# 733. Flood Fill(easy)

Easy

602129FavoriteShare

An image is represented by a 2-D array of integers, each integer representing the pixel value of the image (from 0 to 65535).

Given a coordinate (sr, sc) representing the starting pixel (row and column) of the flood fill, and a pixel value newColor, "flood fill" the image.

To perform a "flood fill", consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color as the starting pixel), and so on. Replace the color of all of the aforementioned pixels with the newColor.

At the end, return the modified image.

**Example 1:**

**Input:**

image = [[1,1,1],[1,1,0],[1,0,1]]

sr = 1, sc = 1, newColor = 2

**Output:** [[2,2,2],[2,2,0],[2,0,1]]

**Explanation:**

From the center of the image (with position (sr, sc) = (1, 1)), all pixels connected

by a path of the same color as the starting pixel are colored with the new color.

Note the bottom corner is not colored 2, because it is not 4-directionally connected

to the starting pixel.

**Note:**

 The length of image and image[0] will be in the range [1, 50].

 The given starting pixel will satisfy 0 <= sr < image.length and 0 <= sc < image[0].length.

 The value of each color in image[i][j] and newColor will be an integer in [0, 65535].

# 744. Find Smallest Letter Greater Than Target(easy)

Easy

272401FavoriteShare

Given a list of sorted characters letters containing only lowercase letters, and given a target letter target, find the smallest element in the list that is larger than the given target.

Letters also wrap around. For example, if the target is target = 'z' and letters = ['a', 'b'], the answer is 'a'.

**Examples:**

**Input:**

letters = ["c", "f", "j"]

target = "a"

**Output:** "c"

**Input:**

letters = ["c", "f", "j"]

target = "c"

**Output:** "f"

**Input:**

letters = ["c", "f", "j"]

target = "d"

**Output:** "f"

**Input:**

letters = ["c", "f", "j"]

target = "g"

**Output:** "j"

**Input:**

letters = ["c", "f", "j"]

target = "j"

**Output:** "c"

**Input:**

letters = ["c", "f", "j"]

target = "k"

**Output:** "c"

**Note:**

1. letters has a length in range [2, 10000].
2. letters consists of lowercase letters, and contains at least 2 unique letters.
3. target is a lowercase letter.

# 746. Min Cost Climbing Stairs(easy)

Easy

1315304FavoriteShare

On a staircase, the i-th step has some non-negative cost cost[i] assigned (0 indexed).

Once you pay the cost, you can either climb one or two steps. You need to find minimum cost to reach the top of the floor, and you can either start from the step with index 0, or the step with index 1.

**Example 1:**

**Input:** cost = [10, 15, 20]

**Output:** 15

**Explanation:** Cheapest is start on cost[1], pay that cost and go to the top.

**Example 2:**

**Input:** cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]

**Output:** 6

**Explanation:** Cheapest is start on cost[0], and only step on 1s, skipping cost[3].

**Note:**

1. cost will have a length in the range [2, 1000].
2. Every cost[i] will be an integer in the range [0, 999].

# 747. Largest Number At Least Twice of Others(easy)

Easy

231462FavoriteShare

In a given integer array nums, there is always exactly one largest element.

Find whether the largest element in the array is at least twice as much as every other number in the array.

If it is, return the **index** of the largest element, otherwise return -1.

**Example 1:**

**Input:** nums = [3, 6, 1, 0]

**Output:** 1

**Explanation:** 6 is the largest integer, and for every other number in the array x,

6 is more than twice as big as x. The index of value 6 is 1, so we return 1.

**Example 2:**

**Input:** nums = [1, 2, 3, 4]

**Output:** -1

**Explanation:** 4 isn't at least as big as twice the value of 3, so we return -1.

**Note:**

1. nums will have a length in the range [1, 50].
2. Every nums[i] will be an integer in the range [0, 99].

# 748. Shortest Completing Word(easy)

Easy

139502FavoriteShare

Find the minimum length word from a given dictionary words, which has all the letters from the string licensePlate. Such a word is said to *complete* the given string licensePlate

Here, for letters we ignore case. For example, "P" on the licensePlate still matches "p" on the word.

It is guaranteed an answer exists. If there are multiple answers, return the one that occurs first in the array.

The license plate might have the same letter occurring multiple times. For example, given a licensePlate of "PP", the word "pair" does not complete the licensePlate, but the word "supper" does.

**Example 1:**

**Input:** licensePlate = "1s3 PSt", words = ["step", "steps", "stripe", "stepple"]

**Output:** "steps"

**Explanation:** The smallest length word that contains the letters "S", "P", "S", and "T".

Note that the answer is not "step", because the letter "s" must occur in the word twice.

Also note that we ignored case for the purposes of comparing whether a letter exists in the word.

**Example 2:**

**Input:** licensePlate = "1s3 456", words = ["looks", "pest", "stew", "show"]

**Output:** "pest"

**Explanation:** There are 3 smallest length words that contains the letters "s".

We return the one that occurred first.

**Note:**

1. licensePlate will be a string with length in range [1, 7].
2. licensePlate will contain digits, spaces, or letters (uppercase or lowercase).
3. words will have a length in the range [10, 1000].
4. Every words[i] will consist of lowercase letters, and have length in range [1, 15].

# 754. Reach a Number(easy)

Easy

347294FavoriteShare

You are standing at position 0 on an infinite number line. There is a goal at position target.

On each move, you can either go left or right. During the *n*-th move (starting from 1), you take *n* steps.

Return the minimum number of steps required to reach the destination.

**Example 1:**

**Input:** target = 3

**Output:** 2

**Explanation:**

On the first move we step from 0 to 1.

On the second step we step from 1 to 3.

**Example 2:**

**Input:** target = 2

**Output:** 3

**Explanation:**

On the first move we step from 0 to 1.

On the second move we step from 1 to -1.

On the third move we step from -1 to 2.

**Note:**

 target will be a non-zero integer in the range [-10^9, 10^9].

# 762. Prime Number of Set Bits in Binary Representation(easy)

Easy

179269FavoriteShare

Given two integers L and R, find the count of numbers in the range [L, R] (inclusive) having a prime number of set bits in their binary representation.

(Recall that the number of set bits an integer has is the number of 1s present when written in binary. For example, 21 written in binary is 10101 which has 3 set bits. Also, 1 is not a prime.)

**Example 1:**

**Input:** L = 6, R = 10

**Output:** 4

**Explanation:**

6 -> 110 (2 set bits, 2 is prime)

7 -> 111 (3 set bits, 3 is prime)

9 -> 1001 (2 set bits , 2 is prime)

10->1010 (2 set bits , 2 is prime)

**Example 2:**

**Input:** L = 10, R = 15

**Output:** 5

**Explanation:**

10 -> 1010 (2 set bits, 2 is prime)

11 -> 1011 (3 set bits, 3 is prime)

12 -> 1100 (2 set bits, 2 is prime)

13 -> 1101 (3 set bits, 3 is prime)

14 -> 1110 (3 set bits, 3 is prime)

15 -> 1111 (4 set bits, 4 is not prime)

**Note:**

1. L, R will be integers L <= R in the range [1, 10^6].
2. R - L will be at most 10000.

# 766. Toeplitz Matrix(easy)

Easy

72569FavoriteShare

A matrix is *Toeplitz* if every diagonal from top-left to bottom-right has the same element.

Now given an M x N matrix, return True if and only if the matrix is *Toeplitz*.

**Example 1:**

**Input:**

matrix = [

  [1,2,3,4],

  [5,1,2,3],

  [9,5,1,2]

]

**Output:** True

**Explanation:**

In the above grid, the diagonals are:

"[9]", "[5, 5]", "[1, 1, 1]", "[2, 2, 2]", "[3, 3]", "[4]".

In each diagonal all elements are the same, so the answer is True.

**Example 2:**

**Input:**

matrix = [

  [1,2],

  [2,2]

]

**Output:** False

**Explanation:**

The diagonal "[1, 2]" has different elements.

**Note:**

1. matrix will be a 2D array of integers.
2. matrix will have a number of rows and columns in range [1, 20].
3. matrix[i][j] will be integers in range [0, 99].

# 771. Jewels and Stones(easy)

Easy

1649307FavoriteShare

You're given strings J representing the types of stones that are jewels, and S representing the stones you have.  Each character in S is a type of stone you have.  You want to know how many of the stones you have are also jewels.

The letters in J are guaranteed distinct, and all characters in J and S are letters. Letters are case sensitive, so "a" is considered a different type of stone from "A".

**Example 1:**

**Input:** J = "aA", S = "aAAbbbb"

**Output:** 3

**Example 2:**

**Input:** J = "z", S = "ZZ"

**Output:** 0

**Note:**

* S and J will consist of letters and have length at most 50.
* The characters in J are distinct.

# 783. Minimum Distance Between BST Nodes(easy)

Easy

441130FavoriteShare

Given a Binary Search Tree (BST) with the root node root, return the minimum difference between the values of any two different nodes in the tree.

**Example :**

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

**Output:** 1

**Explanation:**

Note that root is a TreeNode object, not an array.

The given tree [4,2,6,1,3,null,null] is represented by the following diagram:

4

/ \

2 6

/ \

1 3

while the minimum difference in this tree is 1, it occurs between node 1 and node 2, also between node 3 and node 2.

**Note:**

1. The size of the BST will be between 2 and 100.
2. The BST is always valid, each node's value is an integer, and each node's value is different.

# 784. Letter Case Permutation(easy)

Easy

80992FavoriteShare

Given a string S, we can transform every letter individually to be lowercase or uppercase to create another string.  Return a list of all possible strings we could create.

**Examples:**

**Input:** S = "a1b2"

**Output:** ["a1b2", "a1B2", "A1b2", "A1B2"]

**Input:** S = "3z4"

**Output:** ["3z4", "3Z4"]

**Input:** S = "12345"

**Output:** ["12345"]

**Note:**

* S will be a string with length between 1 and 12.
* S will consist only of letters or digits.

# 788. Rotated Digits(easy)

Easy

250840FavoriteShare

X is a good number if after rotating each digit individually by 180 degrees, we get a valid number that is different from X.  Each digit must be rotated - we cannot choose to leave it alone.

A number is valid if each digit remains a digit after rotation. 0, 1, and 8 rotate to themselves; 2 and 5 rotate to each other; 6 and 9 rotate to each other, and the rest of the numbers do not rotate to any other number and become invalid.

Now given a positive number N, how many numbers X from 1 to N are good?

**Example:**

**Input:** 10

**Output:** 4

**Explanation:**

There are four good numbers in the range [1, 10] : 2, 5, 6, 9.

Note that 1 and 10 are not good numbers, since they remain unchanged after rotating.

**Note:**

* N  will be in range [1, 10000].

# 796. Rotate String(easy)

Easy

48841FavoriteShare

We are given two strings, A and B.

A *shift on A* consists of taking string A and moving the leftmost character to the rightmost position. For example, if A = 'abcde', then it will be 'bcdea' after one shift on A. Return True if and only if A can become B after some number of shifts on A.

**Example 1:**

**Input:** A = 'abcde', B = 'cdeab'

**Output:** true

**Example 2:**

**Input:** A = 'abcde', B = 'abced'

**Output:** false

**Note:**

* A and B will have length at most 100.

# 804. Unique Morse Code Words(easy)

Easy

513478FavoriteShare

International Morse Code defines a standard encoding where each letter is mapped to a series of dots and dashes, as follows: "a" maps to ".-", "b" maps to "-...", "c" maps to "-.-.", and so on.

For convenience, the full table for the 26 letters of the English alphabet is given below:

[".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."]

Now, given a list of words, each word can be written as a concatenation of the Morse code of each letter. For example, "cba" can be written as "-.-..--...", (which is the concatenation "-.-." + "-..." + ".-"). We'll call such a concatenation, the transformation of a word.

Return the number of different transformations among all words we have.

**Example:**

**Input:** words = ["gin", "zen", "gig", "msg"]

**Output:** 2

**Explanation:**

The transformation of each word is:

"gin" -> "--...-."

"zen" -> "--...-."

"gig" -> "--...--."

"msg" -> "--...--."

There are 2 different transformations, "--...-." and "--...--.".

**Note:**

* The length of words will be at most 100.
* Each words[i] will have length in range [1, 12].
* words[i] will only consist of lowercase letters.

# 806. Number of Lines To Write String(easy)

Easy

184732FavoriteShare

We are to write the letters of a given string S, from left to right into lines. Each line has maximum width 100 units, and if writing a letter would cause the width of the line to exceed 100 units, it is written on the next line. We are given an array widths, an array where widths[0] is the width of 'a', widths[1] is the width of 'b', ..., and widths[25] is the width of 'z'.

Now answer two questions: how many lines have at least one character from S, and what is the width used by the last such line? Return your answer as an integer list of length 2.

**Example :**

**Input:**

widths = [10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10]

S = "abcdefghijklmnopqrstuvwxyz"

**Output:** [3, 60]

**Explanation:**

All letters have the same length of 10. To write all 26 letters,

we need two full lines and one line with 60 units.

**Example :**

**Input:**

widths = [4,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10]

S = "bbbcccdddaaa"

**Output:** [2, 4]

**Explanation:**

All letters except 'a' have the same length of 10, and

"bbbcccdddaa" will cover 9 \* 10 + 2 \* 4 = 98 units.

For the last 'a', it is written on the second line because

there is only 2 units left in the first line.

So the answer is 2 lines, plus 4 units in the second line.

**Note:**

* The length of S will be in the range [1, 1000].
* S will only contain lowercase letters.
* widths is an array of length 26.
* widths[i] will be in the range of [2, 10].

# 811. Subdomain Visit Count(easy)

Easy

335504FavoriteShare

A website domain like "discuss.leetcode.com" consists of various subdomains. At the top level, we have "com", at the next level, we have "leetcode.com", and at the lowest level, "discuss.leetcode.com". When we visit a domain like "discuss.leetcode.com", we will also visit the parent domains "leetcode.com" and "com" implicitly.

Now, call a "count-paired domain" to be a count (representing the number of visits this domain received), followed by a space, followed by the address. An example of a count-paired domain might be "9001 discuss.leetcode.com".

We are given a list cpdomains of count-paired domains. We would like a list of count-paired domains, (in the same format as the input, and in any order), that explicitly counts the number of visits to each subdomain.

**Example 1:**

**Input:**

["9001 discuss.leetcode.com"]

**Output:**

["9001 discuss.leetcode.com", "9001 leetcode.com", "9001 com"]

**Explanation:**

We only have one website domain: "discuss.leetcode.com". As discussed above, the subdomain "leetcode.com" and "com" will also be visited. So they will all be visited 9001 times.

**Example 2:**

**Input:**

["900 google.mail.com", "50 yahoo.com", "1 intel.mail.com", "5 wiki.org"]

**Output:**

["901 mail.com","50 yahoo.com","900 google.mail.com","5 wiki.org","5 org","1 intel.mail.com","951 com"]

**Explanation:**

We will visit "google.mail.com" 900 times, "yahoo.com" 50 times, "intel.mail.com" once and "wiki.org" 5 times. For the subdomains, we will visit "mail.com" 900 + 1 = 901 times, "com" 900 + 50 + 1 = 951 times, and "org" 5 times.

**Notes:**

* The length of cpdomains will not exceed 100.
* The length of each domain name will not exceed 100.
* Each address will have either 1 or 2 "." characters.
* The input count in any count-paired domain will not exceed 10000.
* The answer output can be returned in any order.