Google Leetcode questions

Contents

[388. Longest Absolute File Path 4](#_Toc69889016)

[929. Unique Email Addresses 6](#_Toc69889017)

[904. Fruit Into Baskets 7](#_Toc69889018)

[200. Number of Islands 9](#_Toc69889019)

[56. Merge Intervals 10](#_Toc69889020)

[3. Longest Substring Without Repeating Characters 10](#_Toc69889021)

[42. Trapping Rain Water 11](#_Toc69889022)

[66. Plus One 12](#_Toc69889023)

[4. Median of Two Sorted Arrays 13](#_Toc69889024)

[482. License Key Formatting 14](#_Toc69889025)

[146. LRU Cache 15](#_Toc69889026)

[15. 3Sum 16](#_Toc69889027)

[5. Longest Palindromic Substring 17](#_Toc69889028)

[843. Guess the Word 18](#_Toc69889029)

[681. Next Closest Time 19](#_Toc69889030)

[1007. Minimum Domino Rotations For Equal Row 20](#_Toc69889031)

[399. Evaluate Division 22](#_Toc69889032)

[394. Decode String 23](#_Toc69889033)

[844. Backspace String Compare 24](#_Toc69889034)

[20. Valid Parentheses 25](#_Toc69889035)

[683. K Empty Slots 26](#_Toc69889036)

[23. Merge k Sorted Lists 27](#_Toc69889037)

[222. Count Complete Tree Nodes 28](#_Toc69889038)

[253. Meeting Rooms II 30](#_Toc69889039)

[53. Maximum Subarray 30](#_Toc69889040)

[226. Invert Binary Tree 31](#_Toc69889041)

[489. Robot Room Cleaner 33](#_Toc69889042)

[206. Reverse Linked List 35](#_Toc69889043)

[10. Regular Expression Matching 36](#_Toc69889044)

[**139. Word Break** 37](#_Toc69889045)

[**11. Container With Most Water** 38](#_Toc69889046)

[**17. Letter Combinations of a Phone Number** 40](#_Toc69889047)

[**340. Longest Substring with At Most K Distinct Characters** 41](#_Toc69889048)

[**359. Logger Rate Limiter** 41](#_Toc69889049)

[**346. Moving Average from Data Stream** 43](#_Toc69889050)

[**686. Repeated String Match** 44](#_Toc69889051)

[**76. Minimum Window Substring** 45](#_Toc69889052)

[**299. Bulls and Cows** 46](#_Toc69889053)

[**295. Find Median from Data Stream** 47](#_Toc69889054)

[**54. Spiral Matrix** 49](#_Toc69889055)

[**33. Search in Rotated Sorted Array** 50](#_Toc69889056)

[**31. Next Permutation** 51](#_Toc69889057)

[**159. Longest Substring with At Most Two Distinct Characters** 52](#_Toc69889058)

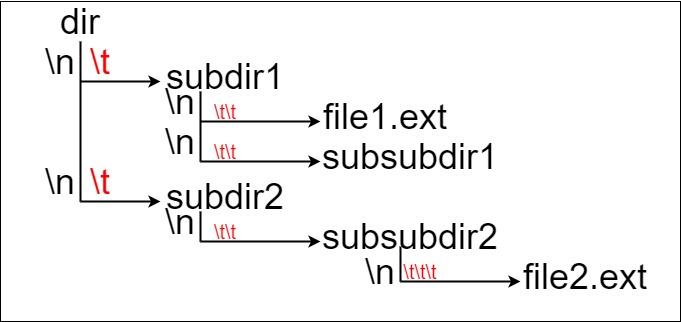
[**163. Missing Ranges** 52](#_Toc69889059)

[**308. Range Sum Query 2D - Mutable** 54](#_Toc69889060)

# [388. Longest Absolute File Path](https://leetcode-cn.com/problems/longest-absolute-file-path/)

难度中等74收藏分享切换为中文接收动态反馈

Suppose we have a file system that stores both files and directories. An example of one system is represented in the following picture:



Here, we have dir as the only directory in the root. dir contains two subdirectories, subdir1 and subdir2. subdir1 contains a file file1.ext and subdirectory subsubdir1. subdir2 contains a subdirectory subsubdir2, which contains a file file2.ext.

In text form, it looks like this (with ⟶ representing the tab character):

dir

⟶ subdir1

⟶ ⟶ file1.ext

⟶ ⟶ subsubdir1

⟶ subdir2

⟶ ⟶ subsubdir2

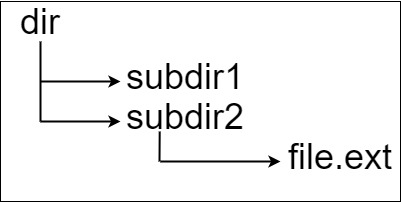
⟶ ⟶ ⟶ file2.ext

If we were to write this representation in code, it will look like this: "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext". Note that the '\n' and '\t' are the new-line and tab characters.

Every file and directory has a unique **absolute path** in the file system, which is the order of directories that must be opened to reach the file/directory itself, all concatenated by '/'s. Using the above example, the **absolute path** to file2.ext is "dir/subdir2/subsubdir2/file2.ext". Each directory name consists of letters, digits, and/or spaces. Each file name is of the form name.extension, where name and extension consist of letters, digits, and/or spaces.

Given a string input representing the file system in the explained format, return *the length of the****longest absolute path****to a****file****in the abstracted file system*. If there is no file in the system, return 0.

**Example 1:**

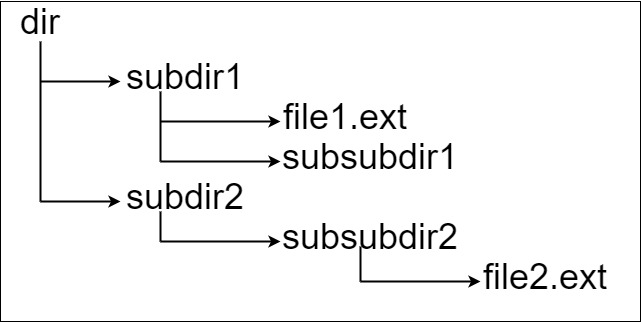


**Input:** input = "dir\n\tsubdir1\n\tsubdir2\n\t\tfile.ext"

**Output:** 20

**Explanation:** We have only one file, and the absolute path is "dir/subdir2/file.ext" of length 20.

**Example 2:**



**Input:** input = "dir\n\tsubdir1\n\t\tfile1.ext\n\t\tsubsubdir1\n\tsubdir2\n\t\tsubsubdir2\n\t\t\tfile2.ext"

**Output:** 32

**Explanation:** We have two files:

"dir/subdir1/file1.ext" of length 21

"dir/subdir2/subsubdir2/file2.ext" of length 32.

We return 32 since it is the longest absolute path to a file.

**Example 3:**

**Input:** input = "a"

**Output:** 0

**Explanation:** We do not have any files, just a single directory named "a".

**Example 4:**

**Input:** input = "file1.txt\nfile2.txt\nlongfile.txt"

**Output:** 12

**Explanation:** There are 3 files at the root directory.

Since the absolute path for anything at the root directory is just the name itself, the answer is "longfile.txt" with length 12.

**Constraints:**

* 1 <= input.length <= 104
* input may contain lowercase or uppercase English letters, a new line character '\n', a tab character '\t', a dot '.', a space ' ', and digits.

通过次数4,594

提交次数8,981

class Solution {

    public int lengthLongestPath(String input) {

    }

}

# [929. Unique Email Addresses](https://leetcode-cn.com/problems/unique-email-addresses/)

难度简单149收藏分享切换为中文接收动态反馈

Every **valid email** consists of a **local name** and a **domain name**, separated by the '@' sign. Besides lowercase letters, the email may contain one or more '.' or '+'.

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

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

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

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

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

It is possible to use both of these rules at the same time.

Given an array of strings emails where we send one email to each email[i], return *the number of different addresses that actually receive mails*.

**Example 1:**

**Input:** emails = ["test.email+alex@leetcode.com","test.e.mail+bob.cathy@leetcode.com","testemail+david@lee.tcode.com"]

**Output:** 2

**Explanation:** "testemail@leetcode.com" and "testemail@lee.tcode.com" actually receive mails.

**Example 2:**

**Input:** emails = ["a@leetcode.com","b@leetcode.com","c@leetcode.com"]

**Output:** 3

**Constraints:**

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

通过次数23,009

提交次数36,001

class Solution {

    public int numUniqueEmails(String[] emails) {

    }

}

# [904. Fruit Into Baskets](https://leetcode-cn.com/problems/fruit-into-baskets/)

难度中等82收藏分享切换为中文接收动态反馈

In a row of trees, the i-th tree produces fruit with type tree[i].

You **start at any tree of your choice**, then repeatedly perform the following steps:

1. Add one piece of fruit from this tree to your baskets.  If you cannot, stop.
2. Move to the next tree to the right of the current tree.  If there is no tree to the right, stop.

Note that you do not have any choice after the initial choice of starting tree: you must perform step 1, then step 2, then back to step 1, then step 2, and so on until you stop.

You have two baskets, and each basket can carry any quantity of fruit, but you want each basket to only carry one type of fruit each.

What is the total amount of fruit you can collect with this procedure?

**Example 1:**

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

**Output:** 3

**Explanation:** We can collect [1,2,1].

**Example 2:**

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

**Output:** 3

**Explanation:** We can collect [1,2,2].

If we started at the first tree, we would only collect [0, 1].

**Example 3:**

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

**Output:** 4

**Explanation:** We can collect [2,3,2,2].

If we started at the first tree, we would only collect [1, 2].

**Example 4:**

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

**Output:** 5

**Explanation:** We can collect [1,2,1,1,2].

If we started at the first tree or the eighth tree, we would only collect 4 fruits.

**Note:**

1. 1 <= tree.length <= 40000
2. 0 <= tree[i] < tree.length

通过次数11,939

提交次数27,088

class Solution {

    public int totalFruit(int[] tree) {

    }

}

# [200. Number of Islands](https://leetcode-cn.com/problems/number-of-islands/)

难度中等1114收藏分享切换为中文接收动态反馈

Given an m x n 2D binary grid grid which represents a map of '1's (land) and '0's (water), return *the number of islands*.

An **island** is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

**Example 1:**

**Input:** grid = [

["1","1","1","1","0"],

["1","1","0","1","0"],

["1","1","0","0","0"],

["0","0","0","0","0"]

]

**Output:** 1

**Example 2:**

**Input:** grid = [

["1","1","0","0","0"],

["1","1","0","0","0"],

["0","0","1","0","0"],

["0","0","0","1","1"]

]

**Output:** 3

**Constraints:**

* m == grid.length
* n == grid[i].length
* 1 <= m, n <= 300
* grid[i][j] is '0' or '1'.

通过次数239,144

提交次数447,402

class Solution {

    public int numIslands(char[][] grid) {

    }

}

# [56. Merge Intervals](https://leetcode-cn.com/problems/merge-intervals/)

难度中等910收藏分享切换为中文接收动态反馈

Given an array of intervals where intervals[i] = [starti, endi], merge all overlapping intervals, and return *an array of the non-overlapping intervals that cover all the intervals in the input*.

**Example 1:**

**Input:** intervals = [[1,3],[2,6],[8,10],[15,18]]

**Output:** [[1,6],[8,10],[15,18]]

**Explanation:** Since intervals [1,3] and [2,6] overlaps, merge them into [1,6].

**Example 2:**

**Input:** intervals = [[1,4],[4,5]]

**Output:** [[1,5]]

**Explanation:** Intervals [1,4] and [4,5] are considered overlapping.

**Constraints:**

* 1 <= intervals.length <= 104
* intervals[i].length == 2
* 0 <= starti <= endi <= 104

通过次数222,111

提交次数490,947

class Solution {

    public int[][] merge(int[][] intervals) {

    }

}

# [3. Longest Substring Without Repeating Characters](https://leetcode-cn.com/problems/longest-substring-without-repeating-characters/)

难度中等5349收藏分享切换为中文接收动态反馈

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

**Example 1:**

**Input:** s = "abcabcbb"

**Output:** 3

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

**Example 2:**

**Input:** s = "bbbbb"

**Output:** 1

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

**Example 3:**

**Input:** s = "pwwkew"

**Output:** 3

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

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

**Example 4:**

**Input:** s = ""

**Output:** 0

**Constraints:**

* 0 <= s.length <= 5 \* 104
* s consists of English letters, digits, symbols and spaces.

通过次数951,683

提交次数2,574,611

class Solution {

    public int lengthOfLongestSubstring(String s) {

    }

}

# [42. Trapping Rain Water](https://leetcode-cn.com/problems/trapping-rain-water/)

难度困难2286收藏分享切换为中文接收动态反馈

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

**Example 1:**



**Input:** height = [0,1,0,2,1,0,1,3,2,1,2,1]

**Output:** 6

**Explanation:** The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

**Example 2:**

**Input:** height = [4,2,0,3,2,5]

**Output:** 9

**Constraints:**

* n == height.length
* 0 <= n <= 3 \* 104
* 0 <= height[i] <= 105

通过次数233,810

提交次数421,558

class Solution {

    public int trap(int[] height) {

    }

}

# [66. Plus One](https://leetcode-cn.com/problems/plus-one/)

难度简单674收藏分享切换为中文接收动态反馈

Given a **non-empty** array of decimal digits representing a non-negative integer, increment 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 contains a single digit.

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

**Example 1:**

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

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

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

**Example 2:**

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

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

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

**Example 3:**

**Input:** digits = [0]

**Output:** [1]

**Constraints:**

* 1 <= digits.length <= 100
* 0 <= digits[i] <= 9

通过次数279,194

提交次数611,137

class Solution {

    public int[] plusOne(int[] digits) {

    }

}

# [4. Median of Two Sorted Arrays](https://leetcode-cn.com/problems/median-of-two-sorted-arrays/)

难度困难4005收藏分享切换为中文接收动态反馈

Given two sorted arrays nums1 and nums2 of size m and n respectively, return **the median** of the two sorted arrays.

**Example 1:**

**Input:** nums1 = [1,3], nums2 = [2]

**Output:** 2.00000

**Explanation:** merged array = [1,2,3] and median is 2.

**Example 2:**

**Input:** nums1 = [1,2], nums2 = [3,4]

**Output:** 2.50000

**Explanation:** merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

**Example 3:**

**Input:** nums1 = [0,0], nums2 = [0,0]

**Output:** 0.00000

**Example 4:**

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

**Output:** 1.00000

**Example 5:**

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

**Output:** 2.00000

**Constraints:**

* nums1.length == m
* nums2.length == n
* 0 <= m <= 1000
* 0 <= n <= 1000
* 1 <= m + n <= 2000
* -106 <= nums1[i], nums2[i] <= 106

**Follow up:** The overall run time complexity should be O(log (m+n)).

通过次数390,075

提交次数975,950

class Solution {

    public double findMedianSortedArrays(int[] nums1, int[] nums2) {

    }

}

# [482. License Key Formatting](https://leetcode-cn.com/problems/license-key-formatting/)

难度简单64收藏分享切换为中文接收动态反馈

You are given a license key represented as a string s that consists of only alphanumeric characters and dashes. The string is separated into n + 1 groups by n dashes. You are also given an integer k.

We want to reformat the string s such that each group contains exactly k characters, except for the first group, which could be shorter than k but still must contain at least one character. Furthermore, there must be a dash inserted between two groups, and you should convert all lowercase letters to uppercase.

Return *the reformatted license key*.

**Example 1:**

**Input:** s = "5F3Z-2e-9-w", k = 4

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

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

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

**Example 2:**

**Input:** s = "2-5g-3-J", k = 2

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

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

**Constraints:**

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

通过次数15,546

提交次数37,247

class Solution {

    public String licenseKeyFormatting(String s, int k) {

    }

}

# [146. LRU Cache](https://leetcode-cn.com/problems/lru-cache/)

难度中等1344收藏分享切换为中文接收动态反馈

Design a data structure that follows the constraints of a [**Least Recently Used (LRU) cache**](https://en.wikipedia.org/wiki/Cache_replacement_policies#LRU).

Implement the LRUCache class:

* LRUCache(int capacity) Initialize the LRU cache with **positive** size capacity.
* int get(int key) Return the value of the key if the key exists, otherwise return -1.
* void put(int key, int value) Update the value of the key if the key exists. Otherwise, add the key-value pair to the cache. If the number of keys exceeds the capacity from this operation, **evict** the least recently used key.

**Follow up:**  
Could you do get and put in O(1) time complexity?

**Example 1:**

**Input**

["LRUCache", "put", "put", "get", "put", "get", "put", "get", "get", "get"]

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

**Output**

[null, null, null, 1, null, -1, null, -1, 3, 4]

**Explanation**

LRUCache lRUCache = new LRUCache(2);

lRUCache.put(1, 1); // cache is {1=1}

lRUCache.put(2, 2); // cache is {1=1, 2=2}

lRUCache.get(1); // return 1

lRUCache.put(3, 3); // LRU key was 2, evicts key 2, cache is {1=1, 3=3}

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

lRUCache.put(4, 4); // LRU key was 1, evicts key 1, cache is {4=4, 3=3}

lRUCache.get(1); // return -1 (not found)

lRUCache.get(3); // return 3

lRUCache.get(4); // return 4

**Constraints:**

* 1 <= capacity <= 3000
* 0 <= key <= 3000
* 0 <= value <= 104
* At most 3 \* 104 calls will be made to get and put.

通过次数161,523

提交次数308,842

class LRUCache {

    public LRUCache(int capacity) {

    }

    public int get(int key) {

    }

    public void put(int key, int value) {

    }

}

/\*\*

 \* Your LRUCache object will be instantiated and called as such:

 \* LRUCache obj = new LRUCache(capacity);

 \* int param\_1 = obj.get(key);

 \* obj.put(key,value);

 \*/

# [15. 3Sum](https://leetcode-cn.com/problems/3sum/)

难度中等3257收藏分享切换为中文接收动态反馈

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.

Notice that the solution set must not contain duplicate triplets.

**Example 1:**

**Input:** nums = [-1,0,1,2,-1,-4]

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

**Example 2:**

**Input:** nums = []

**Output:** []

**Example 3:**

**Input:** nums = [0]

**Output:** []

**Constraints:**

* 0 <= nums.length <= 3000
* -105 <= nums[i] <= 105

通过次数489,039

提交次数1,535,852

class Solution {

    public List<List<Integer>> threeSum(int[] nums) {

    }

}

# [5. Longest Palindromic Substring](https://leetcode-cn.com/problems/longest-palindromic-substring/)

难度中等3547收藏分享切换为中文关闭提醒反馈

Given a string s, return *the longest palindromic substring* in s.

**Example 1:**

**Input:** s = "babad"

**Output:** "bab"

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

**Example 2:**

**Input:** s = "cbbd"

**Output:** "bb"

**Example 3:**

**Input:** s = "a"

**Output:** "a"

**Example 4:**

**Input:** s = "ac"

**Output:** "a"

**Constraints:**

* 1 <= s.length <= 1000
* s consist of only digits and English letters (lower-case and/or upper-case),

通过次数557,655

提交次数1,640,951

class Solution {

    public String longestPalindrome(String s) {

    }

}

# [843. Guess the Word](https://leetcode-cn.com/problems/guess-the-word/)

难度困难76收藏分享切换为中文接收动态反馈

This is an ***interactive problem***.

You are given an array of **unique** strings wordlist where wordlist[i] is 6 letters long, and one word in this list is chosen as secret.

You may call Master.guess(word) to guess a word. The guessed word should have type string and must be from the original list with 6 lowercase letters.

This function returns an integer type, representing the number of exact matches (value and position) of your guess to the secret word. Also, if your guess is not in the given wordlist, it will return -1 instead.

For each test case, you have exactly 10 guesses to guess the word. At the end of any number of calls, if you have made 10 or fewer calls to Master.guess and at least one of these guesses was secret, then you pass the test case.

**Example 1:**

**Input:** secret = "acckzz", wordlist = ["acckzz","ccbazz","eiowzz","abcczz"], numguesses = 10

**Output:** You guessed the secret word correctly.

**Explanation:**

master.guess("aaaaaa") returns -1, because "aaaaaa" is not in wordlist.

master.guess("acckzz") returns 6, because "acckzz" is secret and has all 6 matches.

master.guess("ccbazz") returns 3, because "ccbazz" has 3 matches.

master.guess("eiowzz") returns 2, because "eiowzz" has 2 matches.

master.guess("abcczz") returns 4, because "abcczz" has 4 matches.

We made 5 calls to master.guess and one of them was the secret, so we pass the test case.

**Example 2:**

**Input:** secret = "hamada", wordlist = ["hamada","khaled"], numguesses = 10

**Output:** You guessed the secret word correctly.

**Constraints:**

* 1 <= wordlist.length <= 100
* wordlist[i].length == 6
* wordlist[i] consist of lowercase English letters.
* All the strings of wordlist are **unique**.
* secret exists in wordlist.
* numguesses == 10

通过次数2,176

提交次数5,925

/\*\*

 \* // This is the Master's API interface.

 \* // You should not implement it, or speculate about its implementation

 \* interface Master {

 \*     public int guess(String word) {}

 \* }

 \*/

class Solution {

    public void findSecretWord(String[] wordlist, Master master) {

    }

}

# [681. Next Closest Time](https://leetcode-cn.com/problems/next-closest-time/)

难度中等45收藏分享切换为中文接收动态反馈

Given a time represented in the format "HH:MM", form the next closest time by reusing the current digits. There is no limit on how many times a digit can be reused.

You may assume the given input string is always valid. For example, "01:34", "12:09" are all valid. "1:34", "12:9" are all invalid.

**Example 1:**

**Input:** time = "19:34"

**Output:** "19:39"

**Explanation:** The next closest time choosing from digits **1**, **9**, **3**, **4**, is **19:39**, which occurs 5 minutes later.

It is not **19:33**, because this occurs 23 hours and 59 minutes later.

**Example 2:**

**Input:** time = "23:59"

**Output:** "22:22"

**Explanation:** The next closest time choosing from digits **2**, **3**, **5**, **9**, is **22:22**.

It may be assumed that the returned time is next day's time since it is smaller than the input time numerically.

**Constraints:**

* time.length == 5
* time is a valid time in the form "HH:MM".
* 0 <= HH < 24
* 0 <= MM < 60

通过次数2,109

提交次数4,264

class Solution {

    public String nextClosestTime(String time) {

    }

}

# [1007. Minimum Domino Rotations For Equal Row](https://leetcode-cn.com/problems/minimum-domino-rotations-for-equal-row/)

难度中等72收藏分享切换为中文接收动态反馈

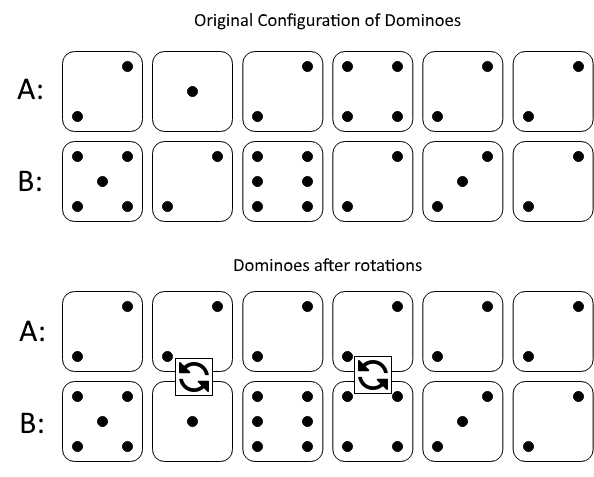
In a row of dominoes, A[i] and B[i] represent the top and bottom halves of the ith domino.  (A domino is a tile with two numbers from 1 to 6 - one on each half of the tile.)

We may rotate the ith domino, so that A[i] and B[i] swap values.

Return the minimum number of rotations so that all the values in A are the same, or all the values in B are the same.

If it cannot be done, return -1.

**Example 1:**



**Input:** A = [2,1,2,4,2,2], B = [5,2,6,2,3,2]

**Output:** 2

**Explanation:**

The first figure represents the dominoes as given by A and B: before we do any rotations.

If we rotate the second and fourth dominoes, we can make every value in the top row equal to 2, as indicated by the second figure.

**Example 2:**

**Input:** A = [3,5,1,2,3], B = [3,6,3,3,4]

**Output:** -1

**Explanation:**

In this case, it is not possible to rotate the dominoes to make one row of values equal.

**Constraints:**

* 2 <= A.length == B.length <= 2 \* 104
* 1 <= A[i], B[i] <= 6

通过次数5,486

提交次数11,981

class Solution {

    public int minDominoRotations(int[] A, int[] B) {

    }

}

# [399. Evaluate Division](https://leetcode-cn.com/problems/evaluate-division/)

难度中等506收藏分享切换为中文接收动态反馈

You are given an array of variable pairs equations and an array of real numbers values, where equations[i] = [Ai, Bi] and values[i] represent the equation Ai / Bi = values[i]. Each Ai or Bi is a string that represents a single variable.

You are also given some queries, where queries[j] = [Cj, Dj] represents the jth query where you must find the answer for Cj / Dj = ?.

Return *the answers to all queries*. If a single answer cannot be determined, return -1.0.

**Note:** The input is always valid. You may assume that evaluating the queries will not result in division by zero and that there is no contradiction.

**Example 1:**

**Input:** equations = [["a","b"],["b","c"]], values = [2.0,3.0], queries = [["a","c"],["b","a"],["a","e"],["a","a"],["x","x"]]

**Output:** [6.00000,0.50000,-1.00000,1.00000,-1.00000]

**Explanation:**

Given: *a / b = 2.0*, *b / c = 3.0*

queries are: *a / c = ?*, *b / a = ?*, *a / e = ?*, *a / a = ?*, *x / x = ?*

return: [6.0, 0.5, -1.0, 1.0, -1.0 ]

**Example 2:**

**Input:** equations = [["a","b"],["b","c"],["bc","cd"]], values = [1.5,2.5,5.0], queries = [["a","c"],["c","b"],["bc","cd"],["cd","bc"]]

**Output:** [3.75000,0.40000,5.00000,0.20000]

**Example 3:**

**Input:** equations = [["a","b"]], values = [0.5], queries = [["a","b"],["b","a"],["a","c"],["x","y"]]

**Output:** [0.50000,2.00000,-1.00000,-1.00000]

**Constraints:**

* 1 <= equations.length <= 20
* equations[i].length == 2
* 1 <= Ai.length, Bi.length <= 5
* values.length == equations.length
* 0.0 < values[i] <= 20.0
* 1 <= queries.length <= 20
* queries[i].length == 2
* 1 <= Cj.length, Dj.length <= 5
* Ai, Bi, Cj, Dj consist of lower case English letters and digits.

通过次数34,771

提交次数58,692

class Solution {

    public double[] calcEquation(List<List<String>> equations, double[] values, List<List<String>> queries) {

    }

}

# [394. Decode String](https://leetcode-cn.com/problems/decode-string/)

难度中等740收藏分享切换为中文接收动态反馈

Given an encoded string, return its decoded string.

The encoding rule is: k[encoded\_string], where the encoded\_string inside the square brackets is being repeated exactly k times. Note that k is guaranteed to be a positive integer.

You may assume that the input string is always valid; No extra white spaces, square brackets are well-formed, etc.

Furthermore, you may assume that the original data does not contain any digits and that digits are only for those repeat numbers, k. For example, there won't be input like 3a or 2[4].

**Example 1:**

**Input:** s = "3[a]2[bc]"

**Output:** "aaabcbc"

**Example 2:**

**Input:** s = "3[a2[c]]"

**Output:** "accaccacc"

**Example 3:**

**Input:** s = "2[abc]3[cd]ef"

**Output:** "abcabccdcdcdef"

**Example 4:**

**Input:** s = "abc3[cd]xyz"

**Output:** "abccdcdcdxyz"

**Constraints:**

* 1 <= s.length <= 30
* s consists of lowercase English letters, digits, and square brackets '[]'.
* s is guaranteed to be **a valid** input.
* All the integers in s are in the range [1, 300].

通过次数91,094

提交次数167,247

class Solution {

    public String decodeString(String s) {

    }

}

# [844. Backspace String Compare](https://leetcode-cn.com/problems/backspace-string-compare/)

难度简单278收藏分享切换为中文接收动态反馈

Given two strings s and t, return true *if they are equal when both are typed into empty text editors*. '#' means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

**Example 1:**

**Input:** s = "ab#c", t = "ad#c"

**Output:** true

**Explanation:** Both s and t become "ac".

**Example 2:**

**Input:** s = "ab##", t = "c#d#"

**Output:** true

**Explanation:** Both s and t become "".

**Example 3:**

**Input:** s = "a##c", t = "#a#c"

**Output:** true

**Explanation:** Both s and t become "c".

**Example 4:**

**Input:** s = "a#c", t = "b"

**Output:** false

**Explanation:** s becomes "c" while t becomes "b".

**Constraints:**

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

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

通过次数69,901

提交次数134,284

class Solution {

    public boolean backspaceCompare(String s, String t) {

    }

}

# [20. Valid Parentheses](https://leetcode-cn.com/problems/valid-parentheses/)

难度简单2344收藏分享切换为中文接收动态反馈

Given a string s 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.

**Example 1:**

**Input:** s = "()"

**Output:** true

**Example 2:**

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

**Output:** true

**Example 3:**

**Input:** s = "(]"

**Output:** false

**Example 4:**

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

**Output:** false

**Example 5:**

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

**Output:** true

**Constraints:**

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

通过次数608,127

提交次数1,382,768

class Solution {

    public boolean isValid(String s) {

    }

}

# [683. K Empty Slots](https://leetcode-cn.com/problems/k-empty-slots/)

难度困难51收藏分享切换为中文接收动态反馈

You have n bulbs in a row numbered from 1 to n. Initially, all the bulbs are turned off. We turn on **exactly one** bulb every day until all bulbs are on after n days.

You are given an array bulbs of length n where bulbs[i] = x means that on the (i+1)th day, we will turn on the bulb at position x where i is **0-indexed** and x is **1-indexed.**

Given an integer k, return *the****minimum day number****such that there exists two****turned on****bulbs that have****exactly****k bulbs between them that are****all turned off****. If there isn't such day, return -1.*

**Example 1:**

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

**Output:** 2

**Explanation:**

On the first day: bulbs[0] = 1, first bulb is turned on: [1,0,0]

On the second day: bulbs[1] = 3, third bulb is turned on: [1,0,1]

On the third day: bulbs[2] = 2, second bulb is turned on: [1,1,1]

We return 2 because on the second day, there were two on bulbs with one off bulb between them.

**Example 2:**

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

**Output:** -1

**Constraints:**

* n == bulbs.length
* 1 <= n <= 2 \* 104
* 1 <= bulbs[i] <= n
* bulbs is a permutation of numbers from 1 to n.
* 0 <= k <= 2 \* 104

通过次数1,771

提交次数4,247

class Solution {

    public int kEmptySlots(int[] bulbs, int k) {

    }

}

# [23. Merge k Sorted Lists](https://leetcode-cn.com/problems/merge-k-sorted-lists/)

难度困难1269收藏分享切换为中文接收动态反馈

You are given an array of k linked-lists lists, each linked-list is sorted in ascending order.

*Merge all the linked-lists into one sorted linked-list and return it.*

**Example 1:**

**Input:** lists = [[1,4,5],[1,3,4],[2,6]]

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

**Explanation:** The linked-lists are:

[

1->4->5,

1->3->4,

2->6

]

merging them into one sorted list:

1->1->2->3->4->4->5->6

**Example 2:**

**Input:** lists = []

**Output:** []

**Example 3:**

**Input:** lists = [[]]

**Output:** []

**Constraints:**

* k == lists.length
* 0 <= k <= 10^4
* 0 <= lists[i].length <= 500
* -10^4 <= lists[i][j] <= 10^4
* lists[i] is sorted in **ascending order**.
* The sum of lists[i].length won't exceed 10^4.

通过次数246,281

提交次数448,674

/\*\*

 \* Definition for singly-linked list.

 \* public class ListNode {

 \*     int val;

 \*     ListNode next;

 \*     ListNode() {}

 \*     ListNode(int val) { this.val = val; }

 \*     ListNode(int val, ListNode next) { this.val = val; this.next = next; }

 \* }

 \*/

class Solution {

    public ListNode mergeKLists(ListNode[] lists) {

    }

}

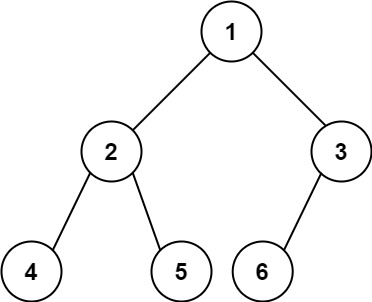
# [222. Count Complete Tree Nodes](https://leetcode-cn.com/problems/count-complete-tree-nodes/)

难度中等474收藏分享切换为中文接收动态反馈

Given the root of a **complete** binary tree, return the number of the nodes in the tree.

According to [**Wikipedia**](http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees), every level, except possibly the last, is completely filled in a complete binary tree, and all nodes in the last level are as far left as possible. It can have between 1 and 2h nodes inclusive at the last level h.

**Example 1:**



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

**Output:** 6

**Example 2:**

**Input:** root = []

**Output:** 0

**Example 3:**

**Input:** root = [1]

**Output:** 1

**Constraints:**

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

**Follow up:** Traversing the tree to count the number of nodes in the tree is an easy solution but with O(n) complexity. Could you find a faster algorithm?

通过次数89,905

提交次数116,229

/\*\*

 \* Definition for a binary tree node.

 \* public class TreeNode {

 \*     int val;

 \*     TreeNode left;

 \*     TreeNode right;

 \*     TreeNode() {}

 \*     TreeNode(int val) { this.val = val; }

 \*     TreeNode(int val, TreeNode left, TreeNode right) {

 \*         this.val = val;

 \*         this.left = left;

 \*         this.right = right;

 \*     }

 \* }

 \*/

class Solution {

    public int countNodes(TreeNode root) {

    }

}

# [253. Meeting Rooms II](https://leetcode-cn.com/problems/meeting-rooms-ii/)

难度中等254收藏分享切换为中文接收动态反馈

Given an array of meeting time intervals intervals where intervals[i] = [starti, endi], return *the minimum number of conference rooms required*.

**Example 1:**

**Input:** intervals = [[0,30],[5,10],[15,20]]

**Output:** 2

**Example 2:**

**Input:** intervals = [[7,10],[2,4]]

**Output:** 1

**Constraints:**

* 1 <= intervals.length <= 104
* 0 <= starti < endi <= 106

通过次数22,927

提交次数47,371

class Solution {

    public int minMeetingRooms(int[][] intervals) {

    }

}

# [53. Maximum Subarray](https://leetcode-cn.com/problems/maximum-subarray/)

难度简单3151收藏分享切换为中文接收动态反馈

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

**Example 1:**

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

**Output:** 6

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

**Example 2:**

**Input:** nums = [1]

**Output:** 1

**Example 3:**

**Input:** nums = [5,4,-1,7,8]

**Output:** 23

**Constraints:**

* 1 <= nums.length <= 3 \* 104
* -105 <= nums[i] <= 105

**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.

通过次数486,164

提交次数900,945

class Solution {

    public int maxSubArray(int[] nums) {

    }

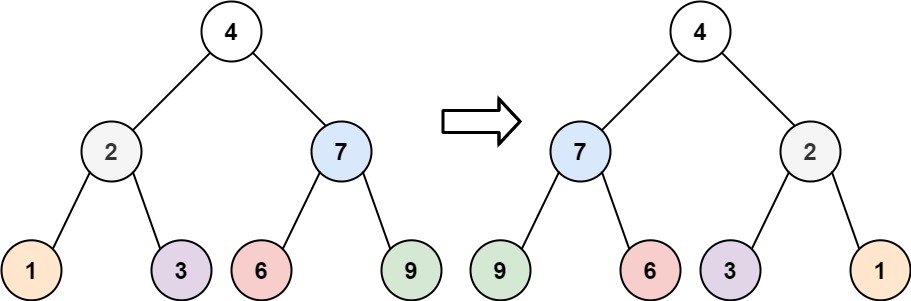
}

# [226. Invert Binary Tree](https://leetcode-cn.com/problems/invert-binary-tree/)

难度简单826收藏分享切换为中文接收动态反馈

Given the root of a binary tree, invert the tree, and return *its root*.

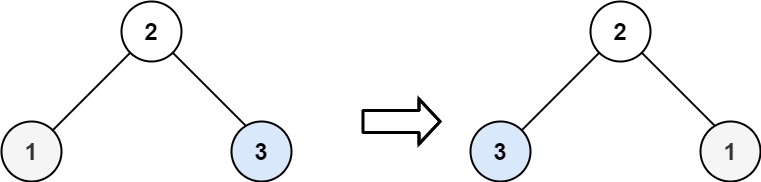
**Example 1:**



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

**Output:** [4,7,2,9,6,3,1]

**Example 2:**



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

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

**Example 3:**

**Input:** root = []

**Output:** []

**Constraints:**

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

通过次数223,629

提交次数285,735

/\*\*

 \* Definition for a binary tree node.

 \* public class TreeNode {

 \*     int val;

 \*     TreeNode left;

 \*     TreeNode right;

 \*     TreeNode() {}

 \*     TreeNode(int val) { this.val = val; }

 \*     TreeNode(int val, TreeNode left, TreeNode right) {

 \*         this.val = val;

 \*         this.left = left;

 \*         this.right = right;

 \*     }

 \* }

 \*/

class Solution {

    public TreeNode invertTree(TreeNode root) {

    }

}

# [489. Robot Room Cleaner](https://leetcode-cn.com/problems/robot-room-cleaner/)

难度困难92收藏分享切换为中文接收动态反馈

Given a robot cleaner in a room modeled as a grid.

Each cell in the grid can be empty or blocked.

The robot cleaner with 4 given APIs can move forward, turn left or turn right. Each turn it made is 90 degrees.

When it tries to move into a blocked cell, its bumper sensor detects the obstacle and it stays on the current cell.

Design an algorithm to clean the entire room using only the 4 given APIs shown below.

interface Robot {

  // returns true if next cell is open and robot moves into the cell.

  // returns false if next cell is obstacle and robot stays on the current cell.

  boolean move();

// Robot will stay on the same cell after calling turnLeft/turnRight.

  // Each turn will be 90 degrees.

  void turnLeft();

  void turnRight();

// Clean the current cell.

void clean();

}

**Example:**

**Input:**

room = [

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

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

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

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

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

],

row = 1,

col = 3

**Explanation:**

All grids in the room are marked by either 0 or 1.

0 means the cell is blocked, while 1 means the cell is accessible.

The robot initially starts at the position of row=1, col=3.

From the top left corner, its position is one row below and three columns right.

**Notes:**

1. The input is only given to initialize the room and the robot's position internally. You must solve this problem "blindfolded". In other words, you must control the robot using only the mentioned 4 APIs, without knowing the room layout and the initial robot's position.
2. The robot's initial position will always be in an accessible cell.
3. The initial direction of the robot will be facing up.
4. All accessible cells are connected, which means the all cells marked as 1 will be accessible by the robot.
5. Assume all four edges of the grid are all surrounded by wall.

通过次数1,276

提交次数1,802

/\*\*

 \* // This is the robot's control interface.

 \* // You should not implement it, or speculate about its implementation

 \* interface Robot {

 \*     // Returns true if the cell in front is open and robot moves into the cell.

 \*     // Returns false if the cell in front is blocked and robot stays in the current cell.

 \*     public boolean move();

 \*

 \*     // Robot will stay in the same cell after calling turnLeft/turnRight.

 \*     // Each turn will be 90 degrees.

 \*     public void turnLeft();

 \*     public void turnRight();

 \*

 \*     // Clean the current cell.

 \*     public void clean();

 \* }

 \*/

class Solution {

    public void cleanRoom(Robot robot) {

    }

}

# [206. Reverse Linked List](https://leetcode-cn.com/problems/reverse-linked-list/)

难度简单1696收藏分享切换为中文接收动态反馈

Given the head of a singly linked list, reverse the list, and return *the reversed list*.

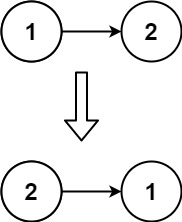
**Example 1:**



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

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

**Example 2:**



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

**Output:** [2,1]

**Example 3:**

**Input:** head = []

**Output:** []

**Constraints:**

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

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

通过次数520,917

提交次数728,005

/\*\*

 \* Definition for singly-linked list.

 \* public class ListNode {

 \*     int val;

 \*     ListNode next;

 \*     ListNode() {}

 \*     ListNode(int val) { this.val = val; }

 \*     ListNode(int val, ListNode next) { this.val = val; this.next = next; }

 \* }

 \*/

class Solution {

    public ListNode reverseList(ListNode head) {

    }

}

# [10. Regular Expression Matching](https://leetcode-cn.com/problems/regular-expression-matching/)

难度困难2057收藏分享切换为中文接收动态反馈

Given an input string (s) and a pattern (p), implement regular expression matching with support for '.' and '\*' where:

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

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

**Example 1:**

**Input:** s = "aa", p = "a"

**Output:** false

**Explanation:** "a" does not match the entire string "aa".

**Example 2:**

**Input:** s = "aa", p = "a\*"

**Output:** true

**Explanation:** '\*' means zero or more of the preceding element, 'a'. Therefore, by repeating 'a' once, it becomes "aa".

**Example 3:**

**Input:** s = "ab", p = ".\*"

**Output:** true

**Explanation:** ".\*" means "zero or more (\*) of any character (.)".

**Example 4:**

**Input:** s = "aab", p = "c\*a\*b"

**Output:** true

**Explanation:** c can be repeated 0 times, a can be repeated 1 time. Therefore, it matches "aab".

**Example 5:**

**Input:** s = "mississippi", p = "mis\*is\*p\*."

**Output:** false

**Constraints:**

* 0 <= s.length <= 20
* 0 <= p.length <= 30
* s contains only lowercase English letters.
* p contains only lowercase English letters, '.', and '\*'.
* It is guaranteed for each appearance of the character '\*', there will be a previous valid character to match.

通过次数162,458

提交次数523,020

class Solution {

    public boolean isMatch(String s, String p) {

    }

}

# [**139. Word Break**](https://leetcode-cn.com/problems/word-break/)

难度中等948收藏分享切换为中文接收动态反馈

Given a string s and a dictionary of strings wordDict, return true if s can be segmented into a space-separated sequence of one or more dictionary words.

**Note** that the same word in the dictionary may be reused multiple times in the segmentation.

**Example 1:**

**Input:** s = "leetcode", wordDict = ["leet","code"]

**Output:** true

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

**Example 2:**

**Input:** s = "applepenapple", wordDict = ["apple","pen"]

**Output:** true

**Explanation:** Return true because "applepenapple" can be segmented as "apple pen apple".

Note that you are allowed to reuse a dictionary word.

**Example 3:**

**Input:** s = "catsandog", wordDict = ["cats","dog","sand","and","cat"]

**Output:** false

**Constraints:**

* 1 <= s.length <= 300
* 1 <= wordDict.length <= 1000
* 1 <= wordDict[i].length <= 20
* s and wordDict[i] consist of only lowercase English letters.
* All the strings of wordDict are **unique**.

通过次数137,096

提交次数275,203

class Solution {

    public boolean wordBreak(String s, List<String> wordDict) {

    }

}

# [**11. Container With Most Water**](https://leetcode-cn.com/problems/container-with-most-water/)

难度中等2389收藏分享切换为中文接收动态反馈

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

**Notice** that you may not slant the container.

**Example 1:**



**Input:** height = [1,8,6,2,5,4,8,3,7]

**Output:** 49

**Explanation:** The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

**Example 2:**

**Input:** height = [1,1]

**Output:** 1

**Example 3:**

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

**Output:** 16

**Example 4:**

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

**Output:** 2

**Constraints:**

* n == height.length
* 2 <= n <= 105
* 0 <= height[i] <= 104

通过次数420,808

提交次数654,863

class Solution {

    public int maxArea(int[] height) {

    }

}

# [**17. Letter Combinations of a Phone Number**](https://leetcode-cn.com/problems/letter-combinations-of-a-phone-number/)

难度中等1263收藏分享切换为中文接收动态反馈

Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in **any order**.

A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



**Example 1:**

**Input:** digits = "23"

**Output:** ["ad","ae","af","bd","be","bf","cd","ce","cf"]

**Example 2:**

**Input:** digits = ""

**Output:** []

**Example 3:**

**Input:** digits = "2"

**Output:** ["a","b","c"]

**Constraints:**

* 0 <= digits.length <= 4
* digits[i] is a digit in the range ['2', '9'].

通过次数255,556

提交次数452,542

class Solution {

    public List<String> letterCombinations(String digits) {

    }

}

# [**340. Longest Substring with At Most K Distinct Characters**](https://leetcode-cn.com/problems/longest-substring-with-at-most-k-distinct-characters/)

难度中等122收藏分享切换为中文接收动态反馈

Given a string s and an integer k, return the length of the longest substring of s that contains at most k ***distinct*** characters.

**Example 1:**

**Input:** s = "eceba", k = 2

**Output:** 3

**Explanation:** The substring is "ece" with length 3.

**Example 2:**

**Input:** s = "aa", k = 1

**Output:** 2

**Explanation:** The substring is "aa" with length 2.

**Constraints:**

* 1 <= s.length <= 5 \* 104
* 0 <= k <= 50

通过次数7,843

提交次数15,904

class Solution {

    public int lengthOfLongestSubstringKDistinct(String s, int k) {

    }

}

# [**359. Logger Rate Limiter**](https://leetcode-cn.com/problems/logger-rate-limiter/)

难度简单42收藏分享切换为中文接收动态反馈

Design a logger system that receives a stream of messages along with their timestamps. Each **unique** message should only be printed **at most every 10 seconds** (i.e. a message printed at timestamp t will prevent other identical messages from being printed until timestamp t + 10).

All messages will come in chronological order. Several messages may arrive at the same timestamp.

Implement the Logger class:

* Logger() Initializes the logger object.
* bool shouldPrintMessage(int timestamp, string message) Returns true if the message should be printed in the given timestamp, otherwise returns false.

**Example 1:**

**Input**

["Logger", "shouldPrintMessage", "shouldPrintMessage", "shouldPrintMessage", "shouldPrintMessage", "shouldPrintMessage", "shouldPrintMessage"]

[[], [1, "foo"], [2, "bar"], [3, "foo"], [8, "bar"], [10, "foo"], [11, "foo"]]

**Output**

[null, true, true, false, false, false, true]

**Explanation**

Logger logger = new Logger();

logger.shouldPrintMessage(1, "foo"); // return true, next allowed timestamp for "foo" is 1 + 10 = 11

logger.shouldPrintMessage(2, "bar"); // return true, next allowed timestamp for "bar" is 2 + 10 = 12

logger.shouldPrintMessage(3, "foo"); // 3 < 11, return false

logger.shouldPrintMessage(8, "bar"); // 8 < 12, return false

logger.shouldPrintMessage(10, "foo"); // 10 < 11, return false

logger.shouldPrintMessage(11, "foo"); // 11 >= 11, return true, next allowed timestamp for "foo" is

// 11 + 10 = 21

**Constraints:**

* 0 <= timestamp <= 109
* Every timestamp will be passed in non-decreasing order (chronological order).
* 1 <= message.length <= 30
* At most 104 calls will be made to shouldPrintMessage.

通过次数3,788

提交次数5,286

class Logger {

    /\*\* Initialize your data structure here. \*/

    public Logger() {

    }

    /\*\* Returns true if the message should be printed in the given timestamp, otherwise returns false.

        If this method returns false, the message will not be printed.

        The timestamp is in seconds granularity. \*/

    public boolean shouldPrintMessage(int timestamp, String message) {

    }

}

/\*\*

 \* Your Logger object will be instantiated and called as such:

 \* Logger obj = new Logger();

 \* boolean param\_1 = obj.shouldPrintMessage(timestamp,message);

 \*/

# [**346. Moving Average from Data Stream**](https://leetcode-cn.com/problems/moving-average-from-data-stream/)

难度简单56收藏分享切换为中文接收动态反馈

Given a stream of integers and a window size, calculate the moving average of all integers in the sliding window.

Implement the MovingAverage class:

* MovingAverage(int size) Initializes the object with the size of the window size.
* double next(int val) Returns the moving average of the last size values of the stream.

**Example 1:**

**Input**

["MovingAverage", "next", "next", "next", "next"]

[[3], [1], [10], [3], [5]]

**Output**

[null, 1.0, 5.5, 4.66667, 6.0]

**Explanation**

MovingAverage movingAverage = new MovingAverage(3);

movingAverage.next(1); // return 1.0 = 1 / 1

movingAverage.next(10); // return 5.5 = (1 + 10) / 2

movingAverage.next(3); // return 4.66667 = (1 + 10 + 3) / 3

movingAverage.next(5); // return 6.0 = (10 + 3 + 5) / 3

**Constraints:**

* 1 <= size <= 1000
* -105 <= val <= 105
* At most 104 calls will be made to next.

通过次数9,523

提交次数13,607

class MovingAverage {

    /\*\* Initialize your data structure here. \*/

    public MovingAverage(int size) {

    }

    public double next(int val) {

    }

}

/\*\*

 \* Your MovingAverage object will be instantiated and called as such:

 \* MovingAverage obj = new MovingAverage(size);

 \* double param\_1 = obj.next(val);

 \*/

# [**686. Repeated String Match**](https://leetcode-cn.com/problems/repeated-string-match/)

难度中等139收藏分享切换为中文接收动态反馈

Given two strings a and b, return the minimum number of times you should repeat string a so that string b is a substring of it. If it is impossible for b​​​​​​ to be a substring of a after repeating it, return -1.

**Notice:** string "abc" repeated 0 times is "",  repeated 1 time is "abc" and repeated 2 times is "abcabc".

**Example 1:**

**Input:** a = "abcd", b = "cdabcdab"

**Output:** 3

**Explanation:** We return 3 because by repeating a three times "ab**cdabcdab**cd", b is a substring of it.

**Example 2:**

**Input:** a = "a", b = "aa"

**Output:** 2

**Example 3:**

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

**Output:** 1

**Example 4:**

**Input:** a = "abc", b = "wxyz"

**Output:** -1

**Constraints:**

* 1 <= a.length <= 104
* 1 <= b.length <= 104
* a and b consist of lower-case English letters.

通过次数15,249

提交次数43,479

class Solution {

    public int repeatedStringMatch(String a, String b) {

    }

}

# [**76. Minimum Window Substring**](https://leetcode-cn.com/problems/minimum-window-substring/)

难度困难1121收藏分享切换为中文接收动态反馈

Given two strings s and t, return the minimum window in *s* which will contain all the characters in *t*. If there is no such window in s that covers all characters in t, return the empty string *""*.

**Note** that If there is such a window, it is guaranteed that there will always be only one unique minimum window in s.

**Example 1:**

**Input:** s = "ADOBECODEBANC", t = "ABC"

**Output:** "BANC"

**Example 2:**

**Input:** s = "a", t = "a"

**Output:** "a"

**Constraints:**

* 1 <= s.length, t.length <= 105
* s and t consist of English letters.

**Follow up:** Could you find an algorithm that runs in O(n) time?

通过次数129,582

提交次数314,417

class Solution {

    public String minWindow(String s, String t) {

    }

}

# [**299. Bulls and Cows**](https://leetcode-cn.com/problems/bulls-and-cows/)

难度中等130收藏分享切换为中文接收动态反馈

You are playing the [**Bulls and Cows**](https://en.wikipedia.org/wiki/Bulls_and_Cows) game with your friend.

You write down a secret number and ask your friend to guess what the number is. When your friend makes a guess, you provide a hint with the following info:

* The number of "bulls", which are digits in the guess that are in the correct position.
* The number of "cows", which are digits in the guess that are in your secret number but are located in the wrong position. Specifically, the non-bull digits in the guess that could be rearranged such that they become bulls.

Given the secret number secret and your friend's guess guess, return the hint for your friend's guess.

The hint should be formatted as "xAyB", where x is the number of bulls and y is the number of cows. Note that both secret and guess may contain duplicate digits.

**Example 1:**

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

**Output:** "1A3B"

**Explanation:** Bulls are connected with a '|' and cows are underlined:

"1807"

|

"7810"

**Example 2:**

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

**Output:** "1A1B"

**Explanation:** Bulls are connected with a '|' and cows are underlined:

"1123" "1123"

| or |

"0111" "0111"

Note that only one of the two unmatched 1s is counted as a cow since the non-bull digits can only be rearranged to allow one 1 to be a bull.

**Example 3:**

**Input:** secret = "1", guess = "0"

**Output:** "0A0B"

**Example 4:**

**Input:** secret = "1", guess = "1"

**Output:** "1A0B"

**Constraints:**

* 1 <= secret.length, guess.length <= 1000
* secret.length == guess.length
* secret and guess consist of digits only.

通过次数25,756

提交次数51,562

class Solution {

    public String getHint(String secret, String guess) {

    }

}

# [**295. Find Median from Data Stream**](https://leetcode-cn.com/problems/find-median-from-data-stream/)

难度困难404收藏分享切换为中文接收动态反馈

The **median** is the middle value in an ordered integer list. If the size of the list is even, there is no middle value and the median is the mean of the two middle values.

* For example, for arr = [2,3,4], the median is 3.
* For example, for arr = [2,3], the median is (2 + 3) / 2 = 2.5.

Implement the MedianFinder class:

* MedianFinder() initializes the MedianFinder object.
* void addNum(int num) adds the integer num from the data stream to the data structure.
* double findMedian() returns the median of all elements so far. Answers within 10-5 of the actual answer will be accepted.

**Example 1:**

**Input**

["MedianFinder", "addNum", "addNum", "findMedian", "addNum", "findMedian"]

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

**Output**

[null, null, null, 1.5, null, 2.0]

**Explanation**

MedianFinder medianFinder = new MedianFinder();

medianFinder.addNum(1); // arr = [1]

medianFinder.addNum(2); // arr = [1, 2]

medianFinder.findMedian(); // return 1.5 (i.e., (1 + 2) / 2)

medianFinder.addNum(3); // arr[1, 2, 3]

medianFinder.findMedian(); // return 2.0

**Constraints:**

* -105 <= num <= 105
* There will be at least one element in the data structure before calling findMedian.
* At most 5 \* 104 calls will be made to addNum and findMedian.

**Follow up:**

* If all integer numbers from the stream are in the range [0, 100], how would you optimize your solution?
* If 99% of all integer numbers from the stream are in the range [0, 100], how would you optimize your solution?

通过次数35,412

提交次数68,406

class MedianFinder {

    /\*\* initialize your data structure here. \*/

    public MedianFinder() {

    }

    public void addNum(int num) {

    }

    public double findMedian() {

    }

}

/\*\*

 \* Your MedianFinder object will be instantiated and called as such:

 \* MedianFinder obj = new MedianFinder();

 \* obj.addNum(num);

 \* double param\_2 = obj.findMedian();

 \*/

# [**54. Spiral Matrix**](https://leetcode-cn.com/problems/spiral-matrix/)

难度中等758收藏分享切换为中文接收动态反馈

Given an m x n matrix, return all elements of the matrix in spiral order.

**Example 1:**



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

**Output:** [1,2,3,6,9,8,7,4,5]

**Example 2:**



**Input:** matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]

**Output:** [1,2,3,4,8,12,11,10,9,5,6,7]

**Constraints:**

* m == matrix.length
* n == matrix[i].length
* 1 <= m, n <= 10
* -100 <= matrix[i][j] <= 100

通过次数147,979

提交次数317,292

class Solution {

    public List<Integer> spiralOrder(int[][] matrix) {

    }

}

# [**33. Search in Rotated Sorted Array**](https://leetcode-cn.com/problems/search-in-rotated-sorted-array/)

难度中等1333收藏分享切换为中文接收动态反馈

There is an integer array nums sorted in ascending order (with **distinct** values).

Prior to being passed to your function, nums is **rotated** at an unknown pivot index k (0 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]] (**0-indexed**). For example, [0,1,2,4,5,6,7] might be rotated at pivot index 3 and become [4,5,6,7,0,1,2].

Given the array nums **after** the rotation and an integer target, return the index of target if it is in nums, or -1 if it is not in nums.

**Example 1:**

**Input:** nums = [4,5,6,7,0,1,2], target = 0

**Output:** 4

**Example 2:**

**Input:** nums = [4,5,6,7,0,1,2], target = 3

**Output:** -1

**Example 3:**

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

**Output:** -1

**Constraints:**

* 1 <= nums.length <= 5000
* -104 <= nums[i] <= 104
* All values of nums are **unique**.
* nums is guaranteed to be rotated at some pivot.
* -104 <= target <= 104

**Follow up:** Can you achieve this in O(log n) time complexity?

通过次数265,861

提交次数641,117

class Solution {

    public int search(int[] nums, int target) {

    }

}

# [**31. Next Permutation**](https://leetcode-cn.com/problems/next-permutation/)

难度中等1071收藏分享切换为中文接收动态反馈

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

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

The replacement must be [**in place**](http://en.wikipedia.org/wiki/In-place_algorithm) and use only constant extra memory.

**Example 1:**

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

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

**Example 2:**

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

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

**Example 3:**

**Input:** nums = [1,1,5]

**Output:** [1,5,1]

**Example 4:**

**Input:** nums = [1]

**Output:** [1]

**Constraints:**

* 1 <= nums.length <= 100
* 0 <= nums[i] <= 100

通过次数156,064

提交次数425,054

class Solution {

    public void nextPermutation(int[] nums) {

    }

}

# [**159. Longest Substring with At Most Two Distinct Characters**](https://leetcode-cn.com/problems/longest-substring-with-at-most-two-distinct-characters/)

难度中等112收藏分享切换为中文接收动态反馈

Given a string s, return the length of the longest substring that contains at most ***two distinct characters***.

**Example 1:**

**Input:** s = "eceba"

**Output:** 3

**Explanation:** The substring is "ece" which its length is 3.

**Example 2:**

**Input:** s = "ccaabbb"

**Output:** 5

**Explanation:** The substring is "aabbb" which its length is 5.

**Constraints:**

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

通过次数10,942

提交次数20,475

class Solution {

    public int lengthOfLongestSubstringTwoDistinct(String s) {

    }

}

# [**163. Missing Ranges**](https://leetcode-cn.com/problems/missing-ranges/)

难度简单40收藏分享切换为中文接收动态反馈

You are given an inclusive range [lower, upper] and a **sorted unique** integer array nums, where all elements are in the inclusive range.

A number x is considered **missing** if x is in the range [lower, upper] and x is not in nums.

Return the ***smallest sorted*** list of ranges that ***cover every missing number exactly***. That is, no element of nums is in any of the ranges, and each missing number is in one of the ranges.

Each range [a,b] in the list should be output as:

* "a->b" if a != b
* "a" if a == b

**Example 1:**

**Input:** nums = [0,1,3,50,75], lower = 0, upper = 99

**Output:** ["2","4->49","51->74","76->99"]

**Explanation:** The ranges are:

[2,2] --> "2"

[4,49] --> "4->49"

[51,74] --> "51->74"

[76,99] --> "76->99"

**Example 2:**

**Input:** nums = [], lower = 1, upper = 1

**Output:** ["1"]

**Explanation:** The only missing range is [1,1], which becomes "1".

**Example 3:**

**Input:** nums = [], lower = -3, upper = -1

**Output:** ["-3->-1"]

**Explanation:** The only missing range is [-3,-1], which becomes "-3->-1".

**Example 4:**

**Input:** nums = [-1], lower = -1, upper = -1

**Output:** []

**Explanation:** There are no missing ranges since there are no missing numbers.

**Example 5:**

**Input:** nums = [-1], lower = -2, upper = -1

**Output:** ["-2"]

**Constraints:**

* -109 <= lower <= upper <= 109
* 0 <= nums.length <= 100
* lower <= nums[i] <= upper
* All the values of nums are **unique**.

通过次数6,648

提交次数21,985

class Solution {

    public List<String> findMissingRanges(int[] nums, int lower, int upper) {

    }

}

# [**308. Range Sum Query 2D - Mutable**](https://leetcode-cn.com/problems/range-sum-query-2d-mutable/)

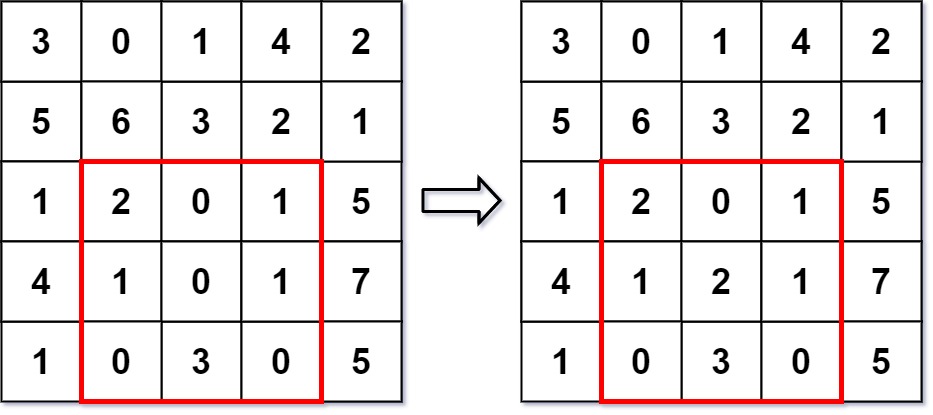
难度困难45收藏分享切换为中文接收动态反馈

Given a 2D matrix matrix, find the sum of the elements inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

Implement the NumMatrix class:

* NumMatrix(int[][] matrix) initializes the object with the integer matrix matrix.
* void update(int row, int col, int val) updates the value of matrix[row][col] to be val.
* int sumRegion(int row1, int col1, int row2, int col2) returns the sum of the elements of the matrix array inside the rectangle defined by its upper left corner (row1, col1) and lower right corner (row2, col2).

**Example 1:**



**Input**

["NumMatrix", "sumRegion", "update", "sumRegion"]

[[[[3, 0, 1, 4, 2], [5, 6, 3, 2, 1], [1, 2, 0, 1, 5], [4, 1, 0, 1, 7], [1, 0, 3, 0, 5]]], [2, 1, 4, 3], [3, 2, 2], [2, 1, 4, 3]]

**Output**

[null, 8, null, 10]

**Explanation**

NumMatrix numMatrix = new NumMatrix([[3, 0, 1, 4, 2], [5, 6, 3, 2, 1], [1, 2, 0, 1, 5], [4, 1, 0, 1, 7], [1, 0, 3, 0, 5]]);

numMatrix.sumRegion(2, 1, 4, 3); // return 8

numMatrix.update(3, 2, 2);

numMatrix.sumRegion(2, 1, 4, 3); // return 10

**Constraints:**

* m == matrix.length
* n == matrix[i].length
* 1 <= m, n <= 200
* -105 <= matrix[i][j] <= 105
* 0 <= row < m
* 0 <= col < n
* -105 <= val <= 105
* 0 <= row1 <= row2 < m
* 0 <= col1 <= col2 < n
* At most 104 calls will be made to sumRegion and update.

通过次数1,740

提交次数2,954

class NumMatrix {

    public NumMatrix(int[][] matrix) {

    }

    public void update(int row, int col, int val) {

    }

    public int sumRegion(int row1, int col1, int row2, int col2) {

    }

}

/\*\*

 \* Your NumMatrix object will be instantiated and called as such:

 \* NumMatrix obj = new NumMatrix(matrix);

 \* obj.update(row,col,val);

 \* int param\_2 = obj.sumRegion(row1,col1,row2,col2);

 \*/