SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(UNIVERSITY ESTABLISHED UNDER SECTION 3 OF UGC ACT 1956)

ENATHUR, KANCHIPURAM – 631 561

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Name: Dhurjati V B K Sai Sushanth

Reg. No: 11249A426

Class: S7 II B.E.(CSE)

Course Name: OOPS LEETCODE PROBLEMS

leetcode.com/problems/majority-element/

Brave isn't your default browser Set as default

Problem List Submit Premium

Description Editorial Solutions Submissions

169. Majority Element

Easy Topics Companies

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

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

Example 2:

```
Input: nums = [2,2,1,1,1,2,2]
Output: 2
```

Constraints:

22K 467 295 Online

Code

C++ Auto

```cpp
class Solution {
public:
    int majorityElement(vector<int>& nums) {
        int count = 0;
        int candidate = 0;
        for (int num : nums) {
            if (count == 0) {
                candidate = num;
```

Saved Ln 14, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =
[3,2,3]

Output

ENG IN 10:33 08/11/2025

leetcode.com/problems/odd-even-linked-list/

Brave isn't your default browser Set as default

Problem List Submit Premium

Description Editorial Solutions Submissions

Run Ctrl '

328. Odd Even Linked List

Medium Topics Companies

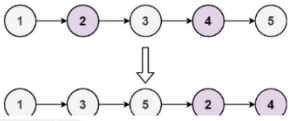
Given the `head` of a singly linked list, group all the nodes with odd indices together followed by the nodes with even indices, and return *the reordered list*.

The **first** node is considered **odd**, and the **second** node is **even**, and so on.

Note that the relative order inside both the even and odd groups should remain as it was in the input.

You must solve the problem in `O(1)` extra space complexity and `O(n)` time complexity.

Example 1:



11.2K 205 64 Online

Code

C++ Auto

```cpp
class Solution {
public:
    ListNode* oddEvenList(ListNode* head) {
        if (!head || !head->next) return head;
        ListNode* odd = head;
        ListNode* even = head->next;
        ListNode* evenHead = even;
        while (even && even->next) {
```

Saved Ln 17, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

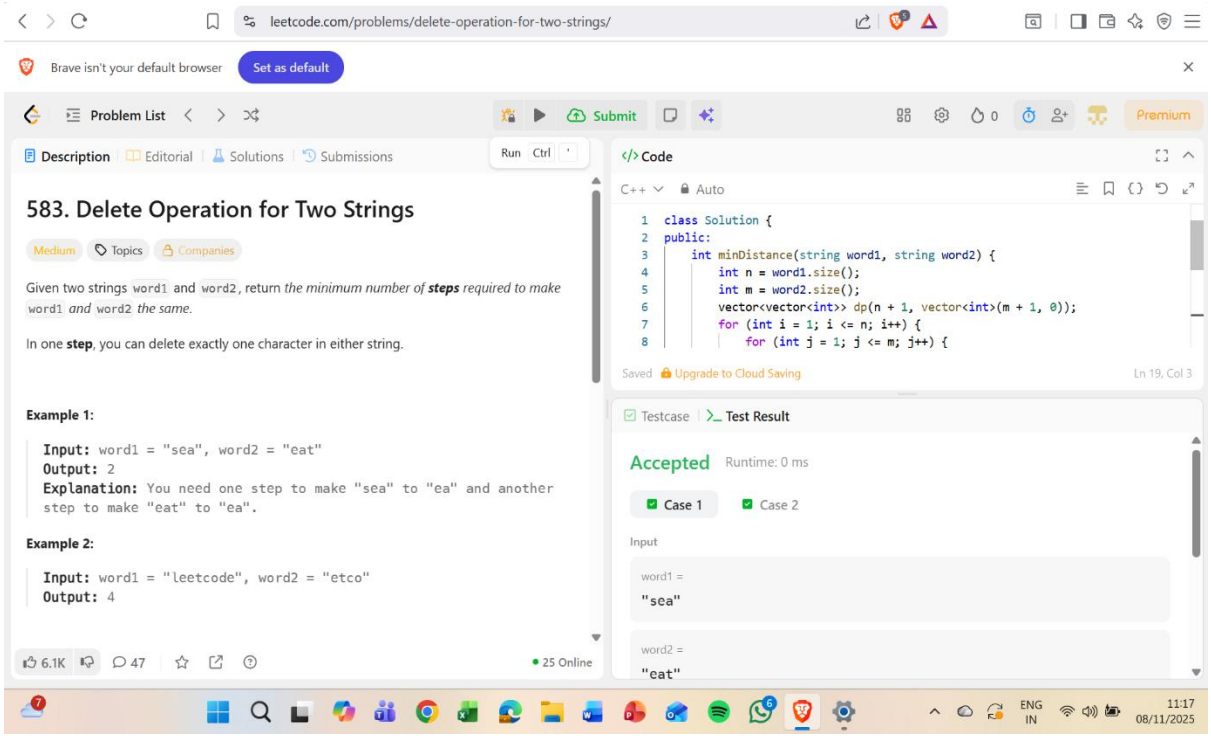
Case 1 Case 2

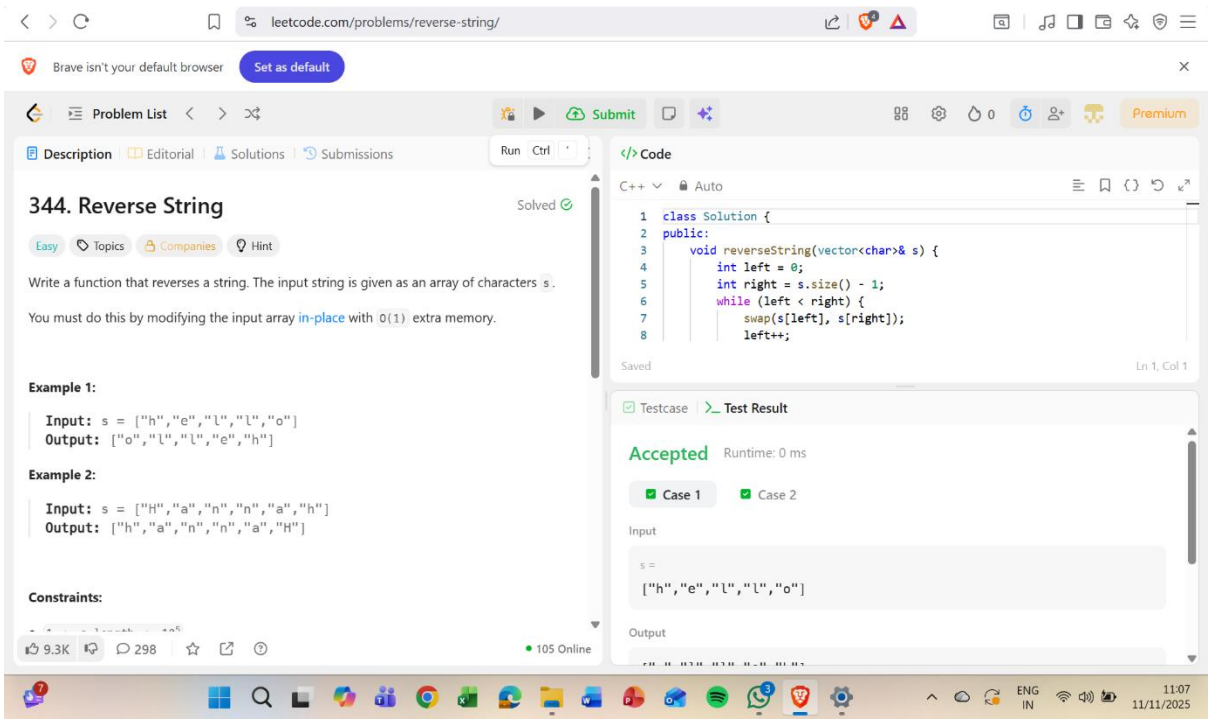
Input

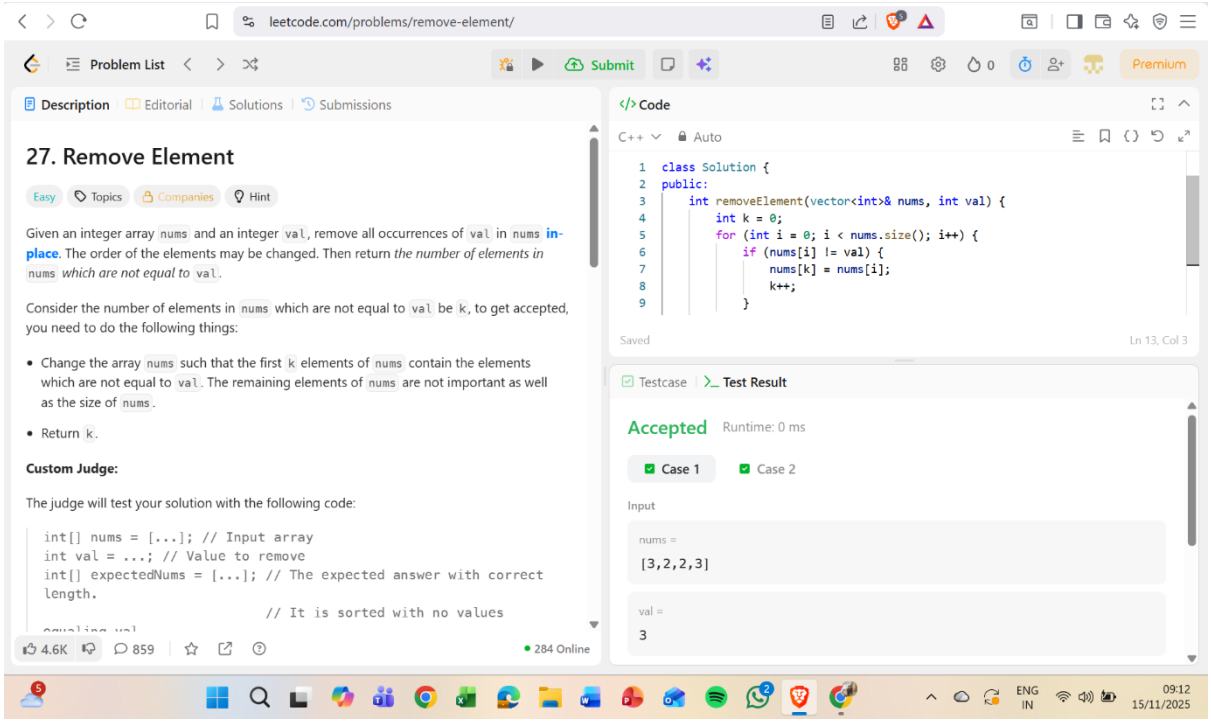
head =
[1,2,3,4,5]

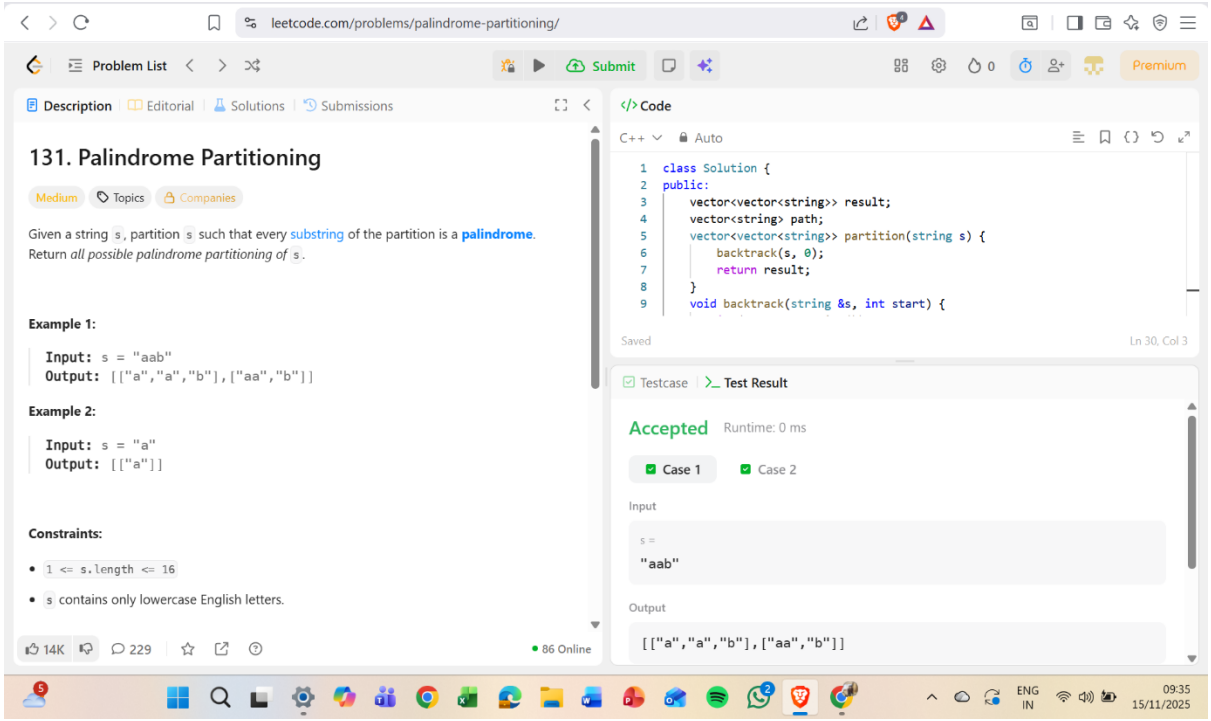
Output

ENG IN 10:48 08/11/2025


leetcode.com/problems/delete-operation-for-two-strings/
Brave isn't your default browser
Set as default
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
Run Ctrl '
583. Delete Operation for Two Strings
Medium
Topics
Companies
Given two strings word1 and word2, return the minimum number of steps required to make word1 and word2 the same.
In one step, you can delete exactly one character in either string.
Example 1:
Input: word1 = "sea", word2 = "eat"
Output: 2
Explanation: You need one step to make "sea" to "ea" and another step to make "eat" to "ea".
Example 2:
Input: word1 = "leetcode", word2 = "etco"
Output: 4
6.1K
47
25 Online
Code
C++
Auto
class Solution {
public:
int minDistance(string word1, string word2) {
int n = word1.size();
int m = word2.size();
vector<vector<int>> dp(n + 1, vector<int>(m + 1, 0));
for (int i = 1; i <= n; i++) {
for (int j = 1; j <= m; j++) {
Saved
Upgrade to Cloud Saving
Ln 19, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
word1 =
"sea"
word2 =
"eat"
ENG
IN
11:17
08/11/2025


leetcode.com/problems/reverse-string/
Brave isn't your default browser
Set as default
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
Run Ctrl '
344. Reverse String
Solved
Easy
Topics
Companies
Hint
Write a function that reverses a string. The input string is given as an array of characters s.
You must do this by modifying the input array in-place with O(1) extra memory.
Example 1:
Input: s = ["h","e","l","l","o"]
Output: ["o","l","l","e","h"]
Example 2:
Input: s = ["H","a","n","n","a","h"]
Output: ["h","a","n","n","a","H"]
Constraints:
9.3K
298
105 Online
Code
C++
Auto
class Solution {
public:
void reverseString(vector<char>& s) {
int left = 0;
int right = s.size() - 1;
while (left < right) {
swap(s[left], s[right]);
left++;
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
s =
["h","e","l","l","o"]
Output
ENG
IN
11:07
11/11/2025


leetcode.com/problems/remove-element/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
27. Remove Element
Easy
Topics
Companies
Hint
Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val.
Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:
Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.
Return k.
Custom Judge:
The judge will test your solution with the following code:
int[] nums = [...]; // Input array
int val = ...; // Value to remove
int[] expectedNums = [...]; // The expected answer with correct length.
// It is sorted with no values
4.6K
859
284 Online
Code
C++
Auto
class Solution {
public:
int removeElement(vector<int>& nums, int val) {
int k = 0;
for (int i = 0; i < nums.size(); i++) {
if (nums[i] != val) {
nums[k] = nums[i];
k++;
}
Saved
Ln 13, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums =
[3,2,2,3]
val =
3
ENG
IN
09:12
15/11/2025


leetcode.com/problems/palindrome-partitioning/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
131. Palindrome Partitioning
Medium
Topics
Companies
Given a string s, partition s such that every substring of the partition is a palindrome. Return all possible palindrome partitioning of s.
Example 1:
Input: s = "aab"
Output: [["a","a","b"],["aa","b"]]
Example 2:
Input: s = "a"
Output: [["a"]]
Constraints:
1 <= s.length <= 16
s contains only lowercase English letters.
14K
229
86 Online
Code
C++
Auto
class Solution {
public:
vector<vector<string>> result;
vector<string> path;
vector<vector<string>> partition(string s) {
backtrack(s, 0);
return result;
}
void backtrack(string &s, int start) {
Saved
Ln 30, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
s =
"aab"
Output
[["a","a","b"],["aa","b"]]
ENG
IN
09:35
15/11/2025

leetcode.com/problems/reverse-linked-list/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

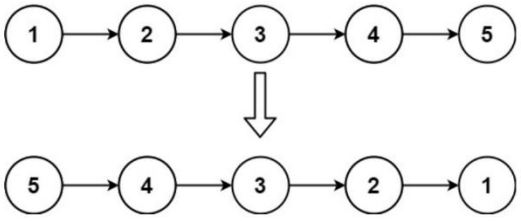
Description | Editorial | Solutions | Submissions

206. Reverse Linked List

Easy | Topics | Companies

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]

23.9K | 362 | 303 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    ListNode* reverseList(ListNode* head) {
        ListNode* prev = nullptr;
        ListNode* curr = head;
        while (curr != nullptr) {
            ListNode* nextNode = curr->next;
            curr->next = prev;
```

Saved | Ln 14, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

head =
[1,2,3,4,5]

Output

ENG IN 10:08 15/11/2025

leetcode.com/problems/rectangle-area/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

223. Rectangle Area

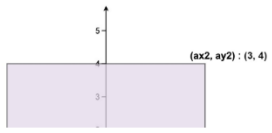
Medium | Topics | Companies

Given the coordinates of two **rectilinear** rectangles in a 2D plane, return *the total area covered by the two rectangles.*

The first rectangle is defined by its **bottom-left** corner `(ax1, ay1)` and its **top-right** corner `(ax2, ay2)`.

The second rectangle is defined by its **bottom-left** corner `(bx1, by1)` and its **top-right** corner `(bx2, by2)`.

Example 1:



2.1K | 45 | 9 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    int computeArea(int ax1, int ay1, int ax2, int ay2,
                    int bx1, int by1, int bx2, int by2) {
        int areaA = (ax2 - ax1) * (ay2 - ay1);
        int areaB = (bx2 - bx1) * (by2 - by1);
        int overlapWidth = max(0, min(ax2, bx2) - max(ax1, bx1));
        int overlapHeight = max(0, min(ay2, by2) - max(ay1, by1));
```

Saved | Ln 12, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

ax1 =
-3

ay1 =
0

ENG IN 10:32 15/11/2025

leetcode.com/problems/intersection-of-two-linked-lists/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

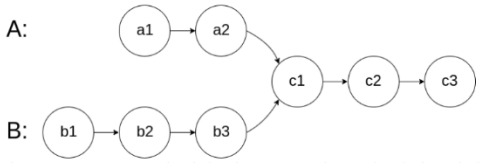
Run Ctrl '

160. Intersection of Two Linked Lists

Easy | Topics | Companies

Given the heads of two singly linked-lists `headA` and `headB`, return *the node at which the two lists intersect*. If the two linked lists have no intersection at all, return `null`.

For example, the following two linked lists begin to intersect at node `c1`:



The test cases are generated such that there are no cycles anywhere in the entire linked structure.

Note that the linked lists must **retain their original structure** after the function returns.

Custom Judge:

16.3K | 256 | 92 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    ListNode *getIntersectionNode(ListNode *headA, ListNode *headB) {
        if (!headA || !headB) return nullptr;
        ListNode* a = headA;
        ListNode* b = headB;
        while (a != b) {
            a = (a == nullptr) ? headB : a->next;
```

Saved | Ln 13, Col 3

Testcase | Test Result

Accepted Runtime: 2 ms

Case 1 | Case 2 | Case 3

Input

intersectVal =
8

listA =
[4,1,8,4,5]

ENG IN 11:26 15/11/2025

leetcode.com/problems/repeated-dna-sequences/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

187. Repeated DNA Sequences

Medium | Topics | Companies

The **DNA sequence** is composed of a series of nucleotides abbreviated as `'A'`, `'C'`, `'G'`, and `'T'`.

- For example, `"ACGAATTCCG"` is a **DNA sequence**.

When studying **DNA**, it is useful to identify repeated sequences within the DNA.

Given a string `s` that represents a **DNA sequence**, return all the `10`-**letter-long** sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in **any order**.

Example 1:

```
Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]
```

Example 2:

3.6K | 81 | 40 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    vector<string> findRepeatedDnaSequences(string s) {
        vector<string> result;
        if (s.length() < 10) return result;
        unordered_set<string> seen;
        unordered_set<string> repeated;
        for (int i = 0; i + 9 < s.length(); i++) {
```

Saved | Ln 9, Col 42

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

s =
"AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"

Output

ENG IN 11:46 15/11/2025

leetcode.com/problems/remove-duplicates-from-sorted-array/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

Run Ctrl '

26. Remove Duplicates from Sorted Array

Easy | Topics | Companies | Hint

Given an integer array `nums` sorted in **non-decreasing order**, remove the duplicates **in-place** such that each unique element appears only **once**. The **relative order** of the elements should be kept the **same**.

Consider the number of *unique elements* in `nums` to be `k`. After removing duplicates, return the number of unique elements `k`.

The first `k` elements of `nums` should contain the unique numbers in **sorted order**. The remaining elements beyond index `k - 1` can be ignored.

Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length

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

18K | 990 | 495 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
        if (nums.empty()) return 0;
        int k = 1;
        for (int i = 1; i < nums.size(); i++) {
            if (nums[i] != nums[i - 1]) {
                nums[k] = nums[i];
```

Saved | Ln 14, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

nums =

[1,1,2]

Output

leetcode.com/problems/remove-nth-node-from-end-of-list/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

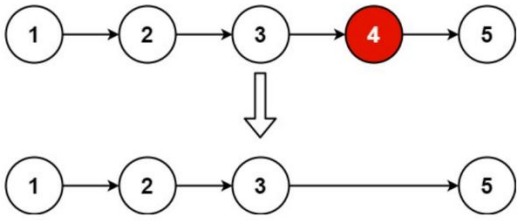
Description | Editorial | Solutions | Submissions

19. Remove Nth Node From End of List

Medium | Topics | Companies | Hint

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

Example 1:



20.8K | 319 | 184 Online

Code

C++ | Auto

```cpp
class Solution {
public:
    ListNode* removeNthFromEnd(ListNode* head, int n) {
        ListNode* dummy = new ListNode(0, head);
        ListNode* fast = dummy;
        ListNode* slow = dummy;
        for (int i = 0; i <= n; i++) {
            fast = fast->next;
```

Saved | Ln 19, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

head =

[1,2,3,4,5]

n =

2

ENG IN 12:03 15/11/2025

ENG IN 12:13 15/11/2025

leetcode.com/problems/remove-duplicates-from-sorted-list-ii/

Brave isn't your default browser | Set as default

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

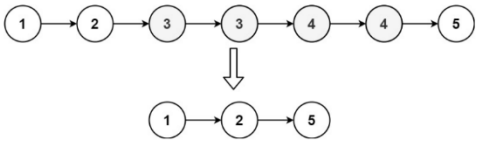
Run Ctrl

82. Remove Duplicates from Sorted List II

Medium | Topics | Companies

Given the `head` of a sorted linked list, *delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list.* Return *the linked list* ***sorted*** *as well.*

Example 1:




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

Example 2:

9.5K | 116 | 53 Online

Code

C++ | Auto

```
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        ListNode* dummy = new ListNode(0, head);
        ListNode* prev = dummy;
        ListNode* curr = head;
        while (curr != nullptr) {
            if (curr->next != nullptr && curr->val == curr->next->val) {
```

Saved | Ln 20, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

head =

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

Output

ENG IN 12:36 15/11/2025

leetcode.com/problems/two-sum/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

1. Two Sum

Easy | Topics | Companies | Hint

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

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

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

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

Example 3:

65.5K | 1.7K | 1823 Online

Code

C++ | Auto

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        unordered_map<int, int> mp;
        for (int i = 0; i < nums.size(); i++) {
            int complement = target - nums[i];
            if (mp.find(complement) != mp.end()) {
                return {mp[complement], i};
            }
```

Saved | Ln 14, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =

[2,7,11,15]

target =

9

ENG IN 11:53 16/11/2025

leetcode.com/problems/add-two-numbers/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

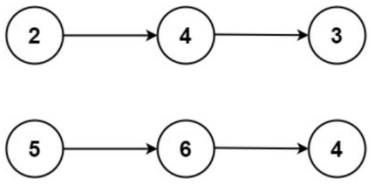
2. Add Two Numbers

Medium | Topics | Companies

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order**, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

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

Example 1:



35.3K | 1K | 751 Online

Code

C++ | Auto

```
class Solution {
public:
    ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
        ListNode* dummy = new ListNode(0);
        ListNode* current = dummy;
        int carry = 0;
        while (l1 != nullptr || l2 != nullptr || carry) {
            int sum = carry;
            if (l1 != nullptr) {
```

Saved | Ln 23, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

l1 =
[2,4,3]

l2 =
[5,6,4]

ENG IN 12:04 16/11/2025

leetcode.com/problems/longest-substring-without-repeating-characters/

Problem List | Submit | Premium

Description | Editorial | Solutions | Submissions

Run Ctrl '

3. Longest Substring Without Repeating Characters

Medium | Topics | Companies | Hint

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

Example 1:

```
Input: s = "abcabcbb"
Output: 3
Explanation: The answer is "abc", with the length of 3. Note that
"bca" and "cab" are also correct answers.
```

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
```

43.6K | 730 | 918 Online

Code

C++ | Auto

```
class Solution {
public:
    int lengthOfLongestSubstring(string s) {
        vector<int> last(256, -1);
        int maxLen = 0, start = 0;
        for (int i = 0; i < s.length(); i++) {
            if (last[s[i]] >= start) {
                start = last[s[i]] + 1;
            }
```

Saved | Ln 15, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

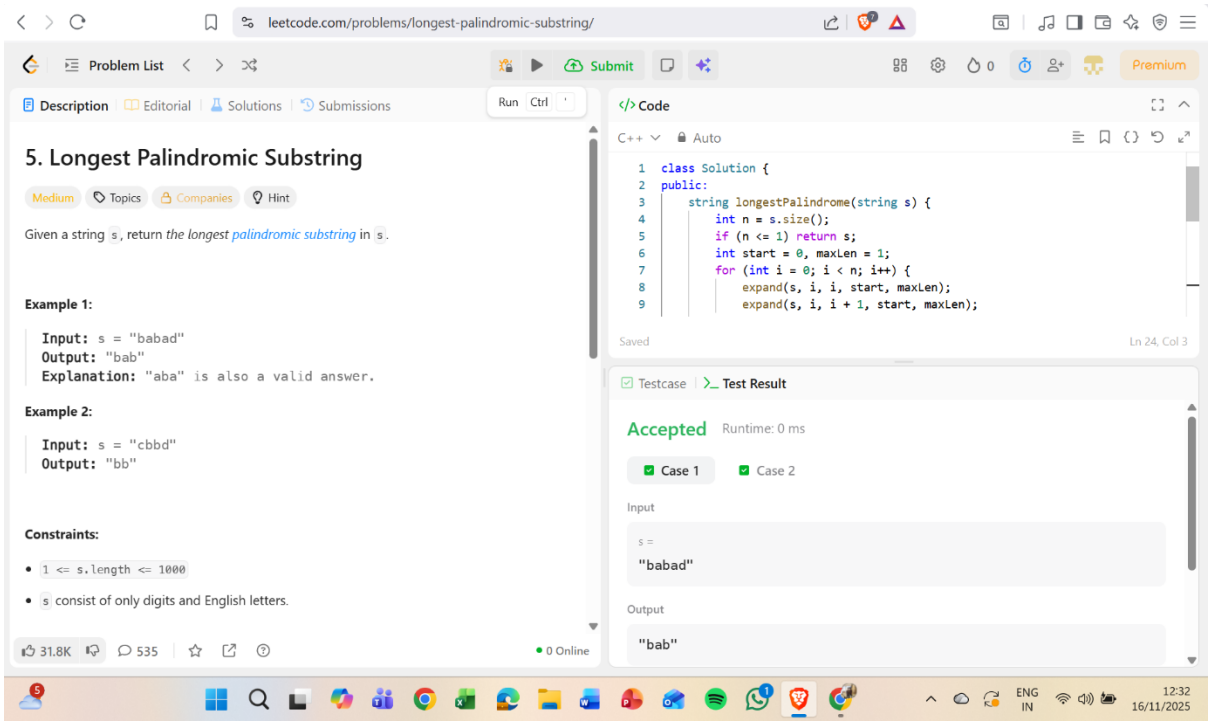
Input

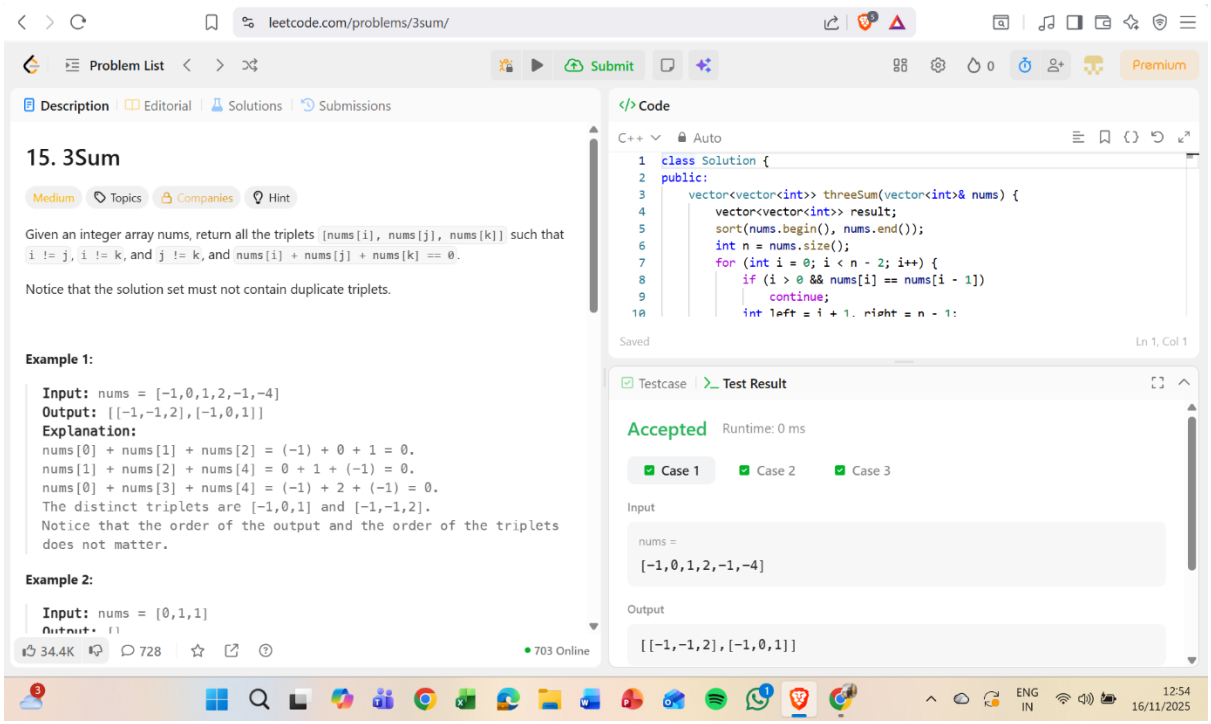
s =
"abcabcbb"

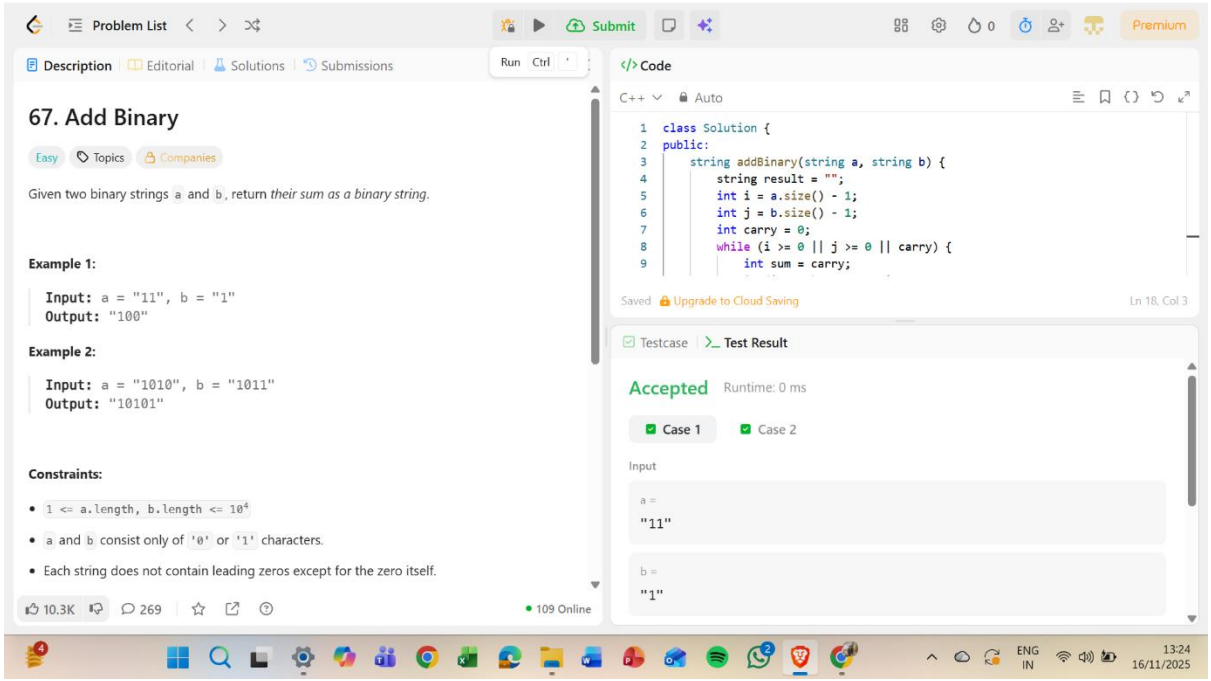
Output

3

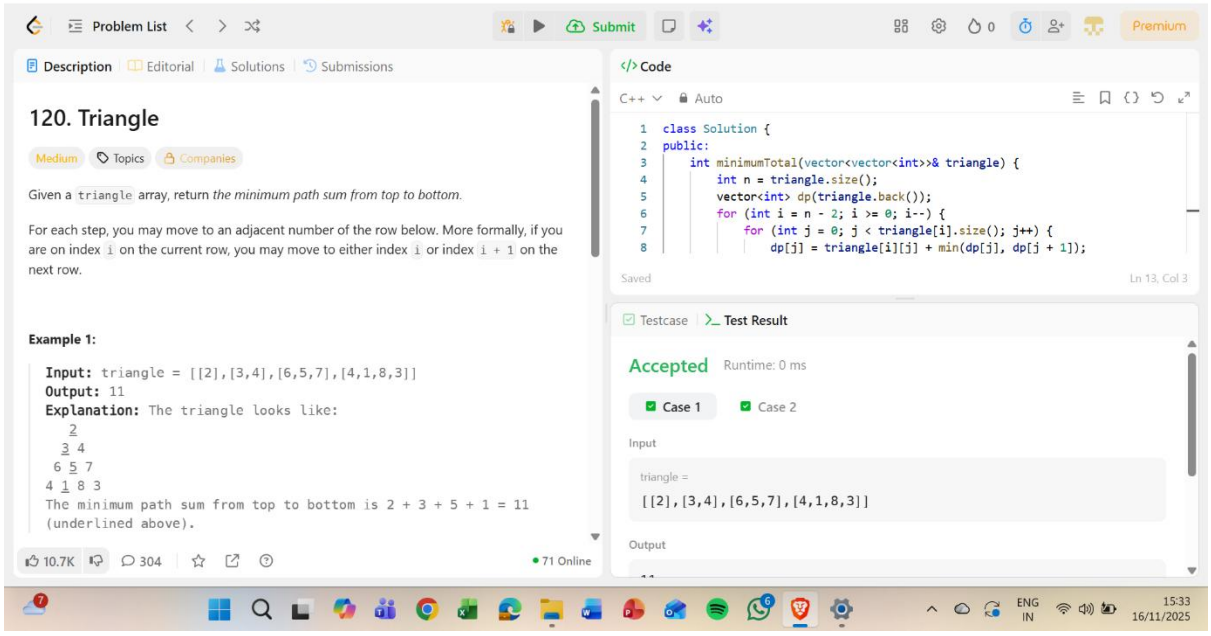
ENG IN 12:19 16/11/2025

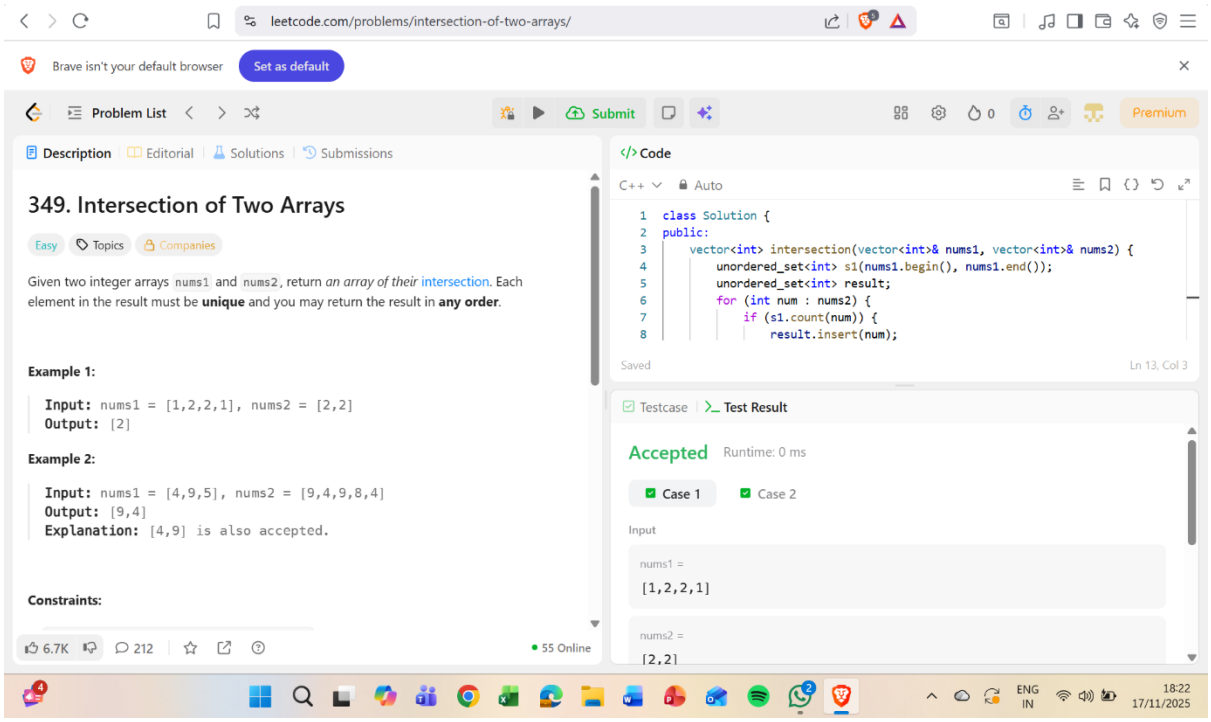

leetcode.com/problems/longest-palindromic-substring/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
Run Ctrl '
5. Longest Palindromic Substring
Medium
Topics
Companies
Hint
Given a string s, return the longest palindromic substring in s.
Example 1:
Input: s = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.
Example 2:
Input: s = "cbbd"
Output: "bb"
Constraints:
1 <= s.length <= 1000
s consist of only digits and English letters.
31.8K
535
0 Online
Code
C++
Auto
class Solution {
public:
string longestPalindrome(string s) {
int n = s.size();
if (n <= 1) return s;
int start = 0, maxLen = 1;
for (int i = 0; i < n; i++) {
expand(s, i, i, start, maxLen);
expand(s, i, i + 1, start, maxLen);
Saved
Ln 24, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
s =
"babad"
Output
"bab"
ENG
IN
12:32
16/11/2025

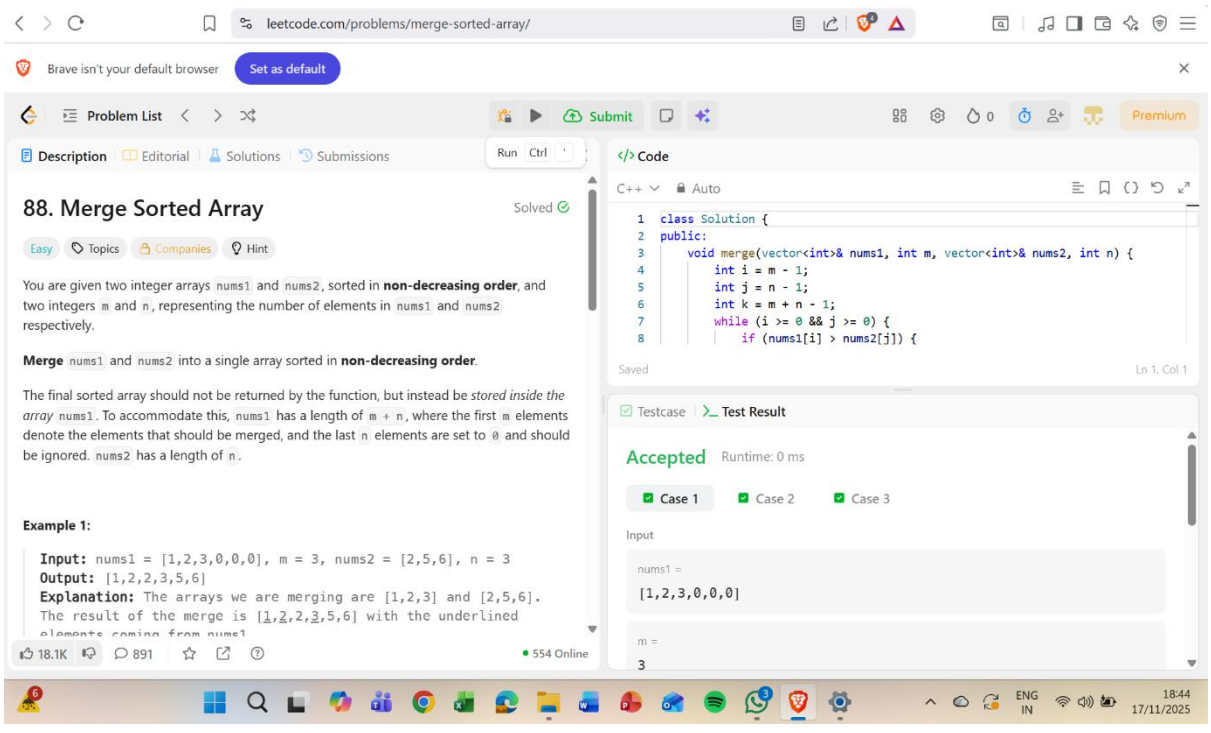

leetcode.com/problems/3sum/
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
15. 3Sum
Medium
Topics
Companies
Hint
Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0.
Notice that the solution set must not contain duplicate triplets.
Example 1:
Input: nums = [-1,0,1,2,-1,-4]
Output: [[-1,-1,2],[-1,0,1]]
Explanation:
nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0.
nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0.
nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0.
The distinct triplets are [-1,0,1] and [-1,-1,2].
Notice that the order of the output and the order of the triplets does not matter.
Example 2:
Input: nums = [0,1,1]
34.4K
728
703 Online
Code
C++
Auto
class Solution {
public:
vector<vector<int>> threeSum(vector<int>& nums) {
vector<vector<int>> result;
sort(nums.begin(), nums.end());
int n = nums.size();
for (int i = 0; i < n - 2; i++) {
if (i > 0 && nums[i] == nums[i - 1])
continue;
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums =
[-1,0,1,2,-1,-4]
Output
[[-1,-1,2],[-1,0,1]]
ENG
IN
12:54
16/11/2025


Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
Run Ctrl '
67. Add Binary
Easy
Topics
Companies
Given two binary strings a and b, return their sum as a binary string.
Example 1:
Input: a = "11", b = "1"
Output: "100"
Example 2:
Input: a = "1010", b = "1011"
Output: "10101"
Constraints:
1 <= a.length, b.length <= 10^4
a and b consist only of '0' or '1' characters.
Each string does not contain leading zeros except for the zero itself.
10.3K
269
109 Online
Code
C++
Auto
class Solution {
public:
string addBinary(string a, string b) {
string result = "";
int i = a.size() - 1;
int j = b.size() - 1;
int carry = 0;
while (i >= 0 || j >= 0 || carry) {
int sum = carry;
Saved
Upgrade to Cloud Saving
Ln 18, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
a =
"11"
b =
"1"


leetcode.com/problems/triangle/
Brave isn't your default browser
Set as default


Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
120. Triangle
Medium
Topics
Companies
Given a triangle array, return the minimum path sum from top to bottom.
For each step, you may move to an adjacent number of the row below. More formally, if you are on index i on the current row, you may move to either index i or index i + 1 on the next row.
Example 1:
Input: triangle = [[2],[3,4],[6,5,7],[4,1,8,3]]
Output: 11
Explanation: The triangle looks like:
2
3 4
6 5 7
4 1 8 3
The minimum path sum from top to bottom is 2 + 3 + 5 + 1 = 11 (underlined above).
10.7K
304
71 Online
Code
C++
Auto
class Solution {
public:
int minimumTotal(vector<vector<int>>& triangle) {
int n = triangle.size();
vector<int> dp(triangle.back());
for (int i = n - 2; i >= 0; i--) {
for (int j = 0; j < triangle[i].size(); j++) {
dp[j] = triangle[i][j] + min(dp[j], dp[j + 1]);
Saved
Ln 13, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
triangle =
[[2],[3,4],[6,5,7],[4,1,8,3]]
Output


leetcode.com/problems/intersection-of-two-arrays/
Brave isn't your default browser
Set as default
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
349. Intersection of Two Arrays
Easy
Topics
Companies
Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must be unique and you may return the result in any order.
Example 1:
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]
Example 2:
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
Explanation: [4,9] is also accepted.
Constraints:
6.7K
212
55 Online
Code
C++
Auto
class Solution {
public:
vector<int> intersection(vector<int>& nums1, vector<int>& nums2) {
unordered_set<int> s1(nums1.begin(), nums1.end());
unordered_set<int> result;
for (int num : nums2) {
if (s1.count(num)) {
result.insert(num);
Saved
Ln 13, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums1 =
[1,2,2,1]
nums2 =
[2,2]
ENG
IN
18:22
17/11/2025


leetcode.com/problems/merge-sorted-array/
Brave isn't your default browser
Set as default
Problem List
Submit
Premium
Run
Ctrl
Description
Editorial
Solutions
Submissions
88. Merge Sorted Array
Solved
Easy
Topics
Companies
Hint
You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.
Merge nums1 and nums2 into a single array sorted in non-decreasing order.
The final sorted array should not be returned by the function, but instead be stored inside the array nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.
Example 1:
Input: nums1 = [1,2,3,0,0,0], m = 3, nums2 = [2,5,6], n = 3
Output: [1,2,2,3,5,6]
Explanation: The arrays we are merging are [1,2,3] and [2,5,6].
The result of the merge is [1,2,2,3,5,6] with the underlined
18.1K
891
554 Online
Code
C++
Auto
class Solution {
public:
void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
int i = m - 1;
int j = n - 1;
int k = m + n - 1;
while (i >= 0 && j >= 0) {
if (nums1[i] > nums2[j]) {
Saved
Ln 1, Col 1
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums1 =
[1,2,3,0,0,0]
m =
3
ENG
IN
18:44
17/11/2025

leetcode.com/problems/same-tree/

Brave isn't your default browser Set as default

Problem List Submit Premium

Description Editorial Solutions Submissions Run Ctrl '

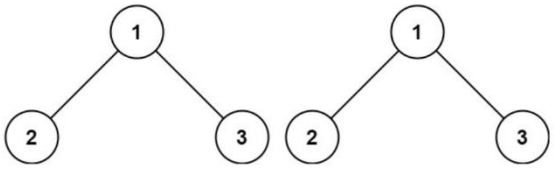
100. Same Tree

Easy Topics Companies

Given the roots of two binary trees `p` and `q`, write a function to check if they are the same or not.

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

Example 1:



12.6K 235 152 Online

Code

C++ Auto

```cpp
class Solution {
public:
    bool isSameTree(TreeNode* p, TreeNode* q) {
        if (!p && !q) return true;
        if (!p || !q || p->val != q->val) return false;
        return isSameTree(p->left, q->left) &&
               isSameTree(p->right, q->right);
    }
```

Saved Ln 9, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

p =
[1,2,3]

q =
[1,2,3]

ENG IN 19:09 17/11/2025

leetcode.com/problems/missing-number/

Brave isn't your default browser Set as default

Problem List Submit Premium

Description Editorial Solutions Submissions Run Ctrl '

268. Missing Number

Easy Topics Companies

Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, return *the only number in the range that is missing from the array.*

Example 1:

> **Input:** `nums = [3,0,1]`
>
> **Output:** 2
>
> **Explanation:**
>
> `n = 3` since there are 3 numbers, so all numbers are in the range `[0,3]`. 2 is the missing number in the range since it does not appear in `nums`.

Example 2:

> **Input:** `nums = [0,1]`

13.8K 361 156 Online

Code

C++ Auto

```cpp
class Solution {
public:
    int missingNumber(vector<int>& nums) {
        int n = nums.size();
        int total = n * (n + 1) / 2;

        int sum = 0;
        for (int x : nums) sum += x;
```

Saved Ln 13, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

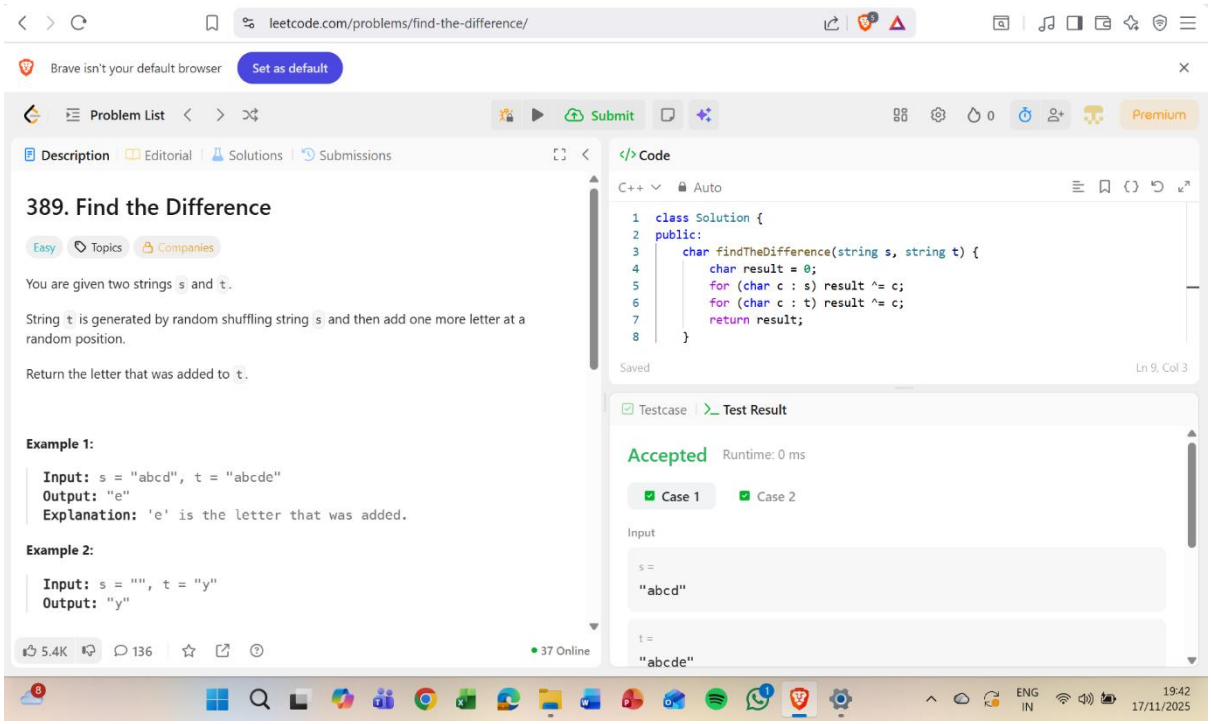
Case 1 Case 2 Case 3

Input

nums =
[3,0,1]

Output

ENG IN 19:23 17/11/2025


leetcode.com/problems/find-the-difference/
Brave isn't your default browser
Set as default
Problem List
Submit
Premium
Description
Editorial
Solutions
Submissions
389. Find the Difference
Easy
Topics
Companies
You are given two strings s and t.
String t is generated by random shuffling string s and then add one more letter at a random position.
Return the letter that was added to t.
Example 1:
Input: s = "abcd", t = "abcde"
Output: "e"
Explanation: 'e' is the letter that was added.
Example 2:
Input: s = "", t = "y"
Output: "y"
5.4K
136
37 Online
Code
C++
Auto
1 class Solution {
2 public:
3 char findTheDifference(string s, string t) {
4 char result = 0;
5 for (char c : s) result ^= c;
6 for (char c : t) result ^= c;
7 return result;
8 }
Saved
Ln 9, Col 3
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
s =
"abcd"
t =
"abcde"
ENG
IN
19:42
17/11/2025