

Screenshots of Submissions

1. Two Sum

Problem List < > ↺ Submit

Editorial Solutions Submissions Accepted × Description

← All Submissions

Accepted 63 / 63 testcases passed
23BCS10849 submitted at Aug 24, 2025 18:07

Editorial Solution

Runtime 0 ms Beats 100.00% 🏆
Memory 14.93 MB Beats 20.34%

Analyze Complexity

Code C++

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
```

</> Code C++ Auto

```
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4         unordered_map<int, int> map;
5
6         for (int i = 0; i < nums.size(); i++) {
7             int complement = target - nums[i];
8
9             if (map.find(complement) != map.end()) {
10
11                 return {map[complement], i};
12             }
13         }
14     }
15 }
```

Saved

Test Result Testcase

Case 1 Case 2 Case 3 +

nums =

[2,7,11,15]

target =

9

</> Source ?

2. Rotate Image

Problem List < > ↺ Submit

Editorial Solutions Submissions Description Accepted ×

← All Submissions

Accepted 21 / 21 testcases passed
23BCS10849 submitted at Aug 24, 2025 18:10

Editorial Solution

Runtime 0 ms Beats 100.00% 🏆
Memory 10.14 MB Beats 68.51% 🏆

Analyze Complexity

Code C++

```
class Solution {
public:
    void rotate(vector<vector<int>>& matrix) {
        int n = matrix.size();
```

</> Code C++ Auto

```
1 class Solution {
2 public:
3     void rotate(vector<vector<int>>& matrix) {
4         int n = matrix.size();
5
6         // Step 1: Transpose the matrix
7         for(int i = 0; i < n; i++) {
8             for(int j = i; j < n; j++) {
9                 swap(matrix[i][j], matrix[j][i]);
10             }
11         }
12     }
13 }
```

Saved

Test Result Testcase

Case 1 Case 2 +

matrix =

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

</> Source ?

3. Next Permutations

The screenshot shows a C++ solution for the "Next Permutations" problem. The code is as follows:

```
1 class Solution {
2 public:
3     void nextPermutation(vector<int>& nums) {
4         int i = nums.size() - 1;
5         while (i > 0 && nums[i-1] >= nums[i]) {
6             i--;
7         }
8
9         if (i == 0) {
10             reverse(nums.begin(), nums.end());
11         }
12     }
13 }
```

The submission status is "Accepted" with 266 / 266 testcases passed. The runtime is 0 ms (Beats 100.00%) and memory is 15.69 MB (Beats 67.60%). A bar chart shows the runtime performance relative to other submissions.

Testcase 1 input: `nums = [1,2,3]`

4. Postorder Traversal

The screenshot shows a C++ solution for the "Binary Tree Postorder Traversal" problem. The code is as follows:

```
13 public:
14     vector<int> postorderTraversal(TreeNode* root) {
15         vector<int> post;
16         if (root == NULL) {
17             return post;
18         }
19         stack<TreeNode*> st1;
20         stack<int> st2;
21         st1.push(root);
22         while (!st1.empty()) {
23             TreeNode* temp = st1.top();
24             // ...
25         }
26     }
27 }
```

The submission status is "Accepted" with 71 / 71 testcases passed. The runtime is 0 ms (Beats 100.00%) and memory is 10.89 MB (Beats 92.29%). A bar chart shows the runtime performance relative to other submissions.

Testcase 1 input: `root = [1,null,2,3]`

Testcase 1 output: `[3,2,1]`

5. Inorder Traversal

Editorial Solutions Submissions Description Accepted X

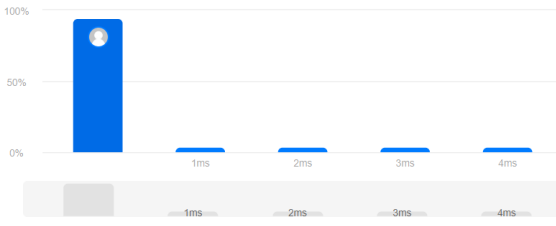
All Submissions

Accepted 71 / 71 testcases passed
23BCS10849 submitted at Aug 24, 2025 18:17

Editorial Solution

Runtime 0 ms Beats 100.00%
Memory 10.75 MB Beats 86.73%

Analyze Complexity



Case	Runtime (ms)
1	0
2	0
3	0
4	0
5	0

Code C++

/**
 * Definition for a binary tree node.
 * struct TreeNode {
 * int val;
 * }

Code

C++ Auto

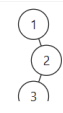
12 class Solution {
13 public:
14 vector<int> inorderTraversal(TreeNode* root) {
15 vector<int> result;
16 inorder(root, result);
17 return result;
18 }
19 private:
20 void inorder(TreeNode* node, vector<int>& result) {
21 if (!node) return;
22 inorder(node->left, result);
23 result.push_back(node->val);
24 inorder(node->right, result);
25 }
26 }

Saved

Test Result Testcase

Case 1 Case 2 Case 3 Case 4 +

root =
[1,null,2,3]



```
graph TD; 1((1)) --> null((null)); 1 --> 2((2)); 2 --> 3((3));
```

</> Source

6. Preorder Traversal

Welcome to Chandigarh University Stack template operations Online C++ Compiler and Visual Studio Binary Tree Preorder Traversal

leetcode.com/problems/binary-tree-preorder-traversal/submissions/1746611476/

Problem List < > Submit

Editorial Solutions Accepted X Submissions Description

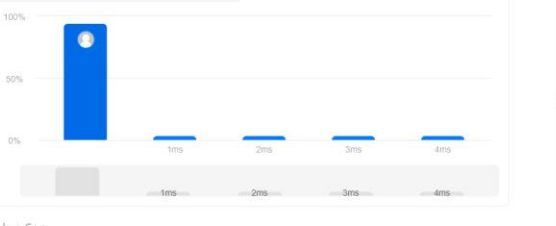
All Submissions

Accepted 71 / 71 testcases passed
23BCS10849 submitted at Aug 24, 2025 18:19

Editorial Solution

Runtime 0 ms Beats 100.00%
Memory 10.93 MB Beats 40.31%

Analyze Complexity



Case	Runtime (ms)
1	0
2	0
3	0
4	0
5	0

Code C++

/**
 * Definition for a binary tree node.
 * struct TreeNode {
 * int val;
 * }

Code

C++ Auto

12 class Solution {
13 public:
14 vector<int> preorderTraversal(TreeNode* root) {
15 vector<int> preorder;
16 if (root == NULL) {
17 return preorder;
18 }
19 stack<TreeNode*> st;
20 st.push(root);
21 while (!st.empty()) {
22 root = st.top();
23 preorder.push_back(root->val);
24 if (root->right) {
25 st.push(root->right);
26 }
27 root = root->left;
28 }
29 return preorder;
30 }
31 }

Saved

Test Result Testcase

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3 Case 4

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

Output
[1,2,3]

29°C Light rain 18:19 24-08-2025

7. Ancestor of the tree

Screenshot of a LeetCode submission for the problem "Lowest Common Ancestor of a Binary Tree".

The submission is for the problem "Lowest Common Ancestor of a Binary Tree" (ID 238CS10849) and is marked as "Accepted". It shows the runtime performance: 11 ms, Beats: 66.40%, and Memory: 17.51 MB, Beats: 17.93%.

The code is written in C++ and implements a recursive solution to find the lowest common ancestor (LCA) of two nodes in a binary tree. The code defines a `TreeNode` structure and a `lowestCommonAncestor` function.

```
class Solution {
public:
    TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
        if (root == nullptr || root == p || root == q) {
            return root;
        }

        TreeNode* left = lowestCommonAncestor(root->left, p, q);
        TreeNode* right = lowestCommonAncestor(root->right, p, q);

        if (left != nullptr && right != nullptr) {
            return root;
        }
    }
};
```

The test result shows the submission is "Accepted" with a runtime of 3 ms. The input for the test case is:

```
root = [3,5,1,6,2,0,8,null,null,7,4]
p = 5
```

The bottom status bar shows the system time as 18:20 on 24-08-2025, with a temperature of 29°C and light rain.