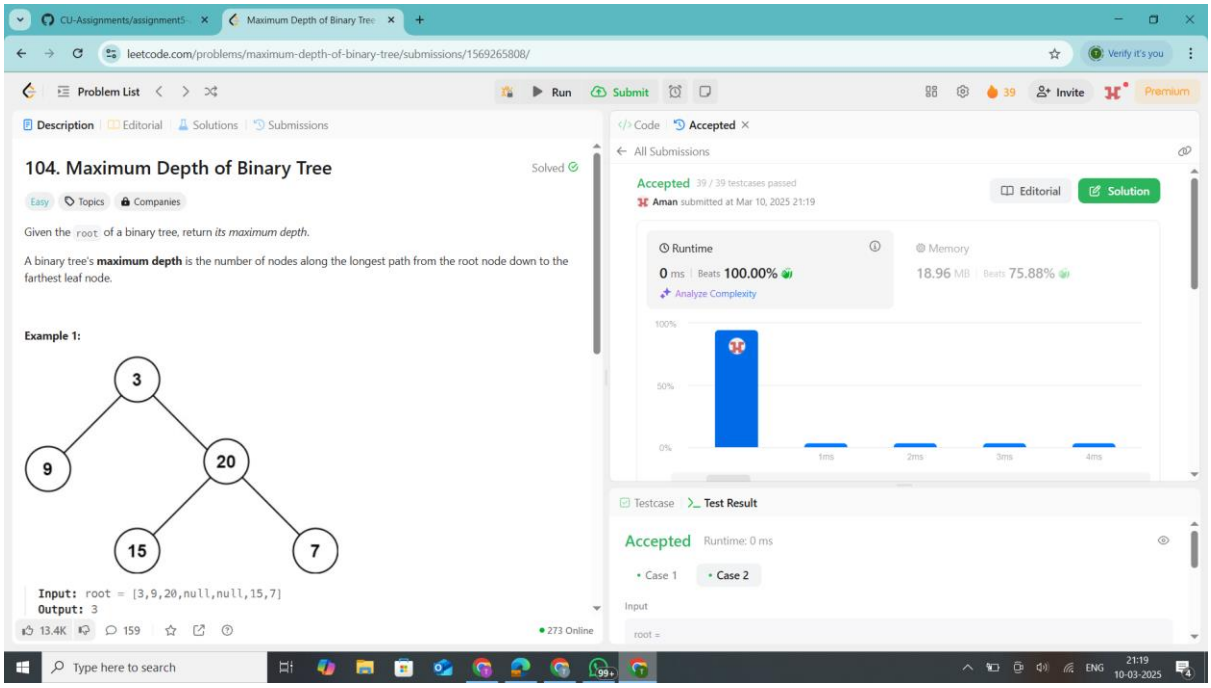
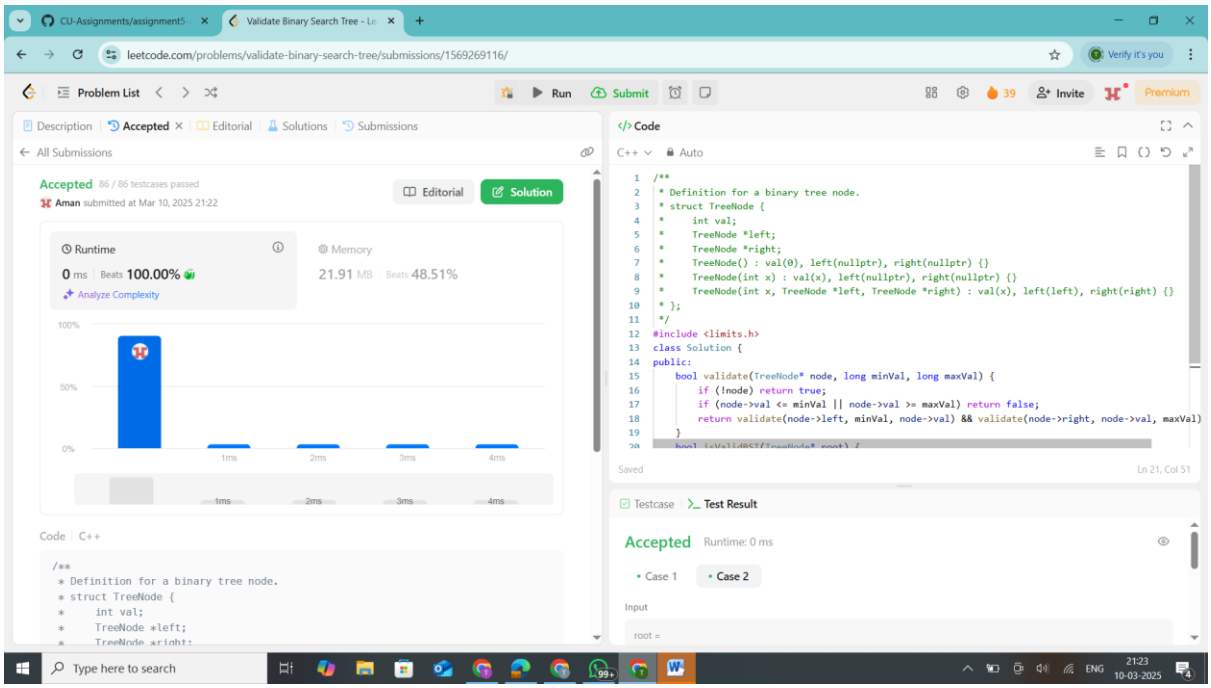
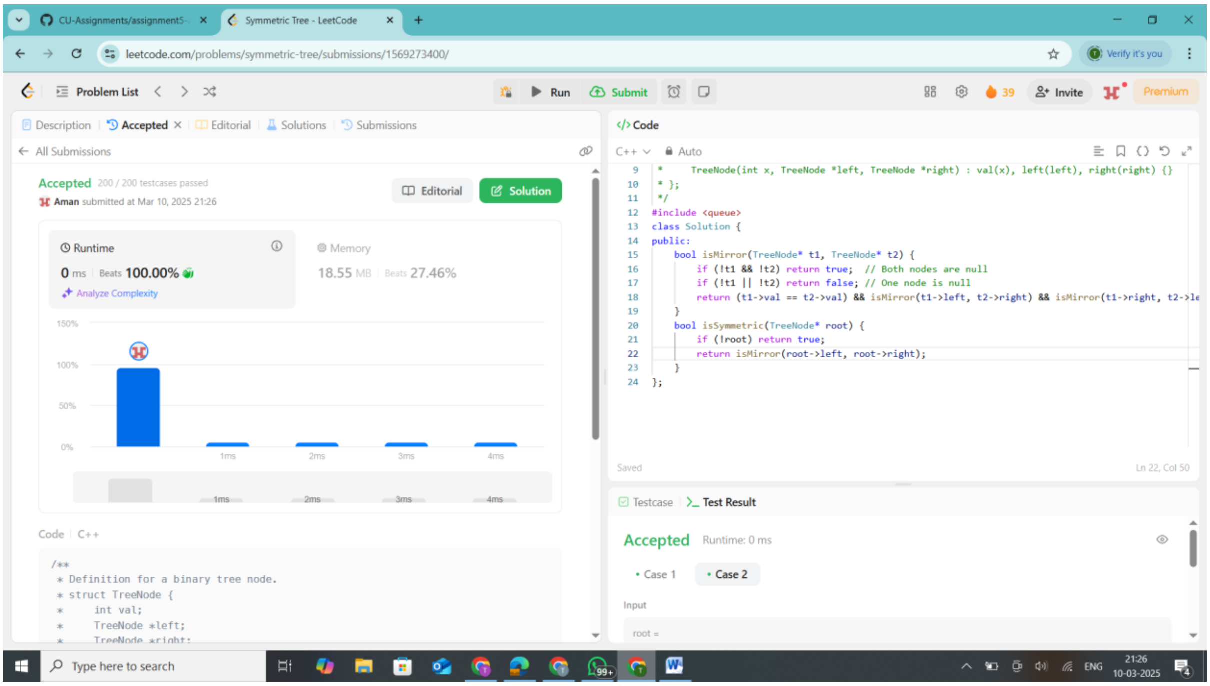
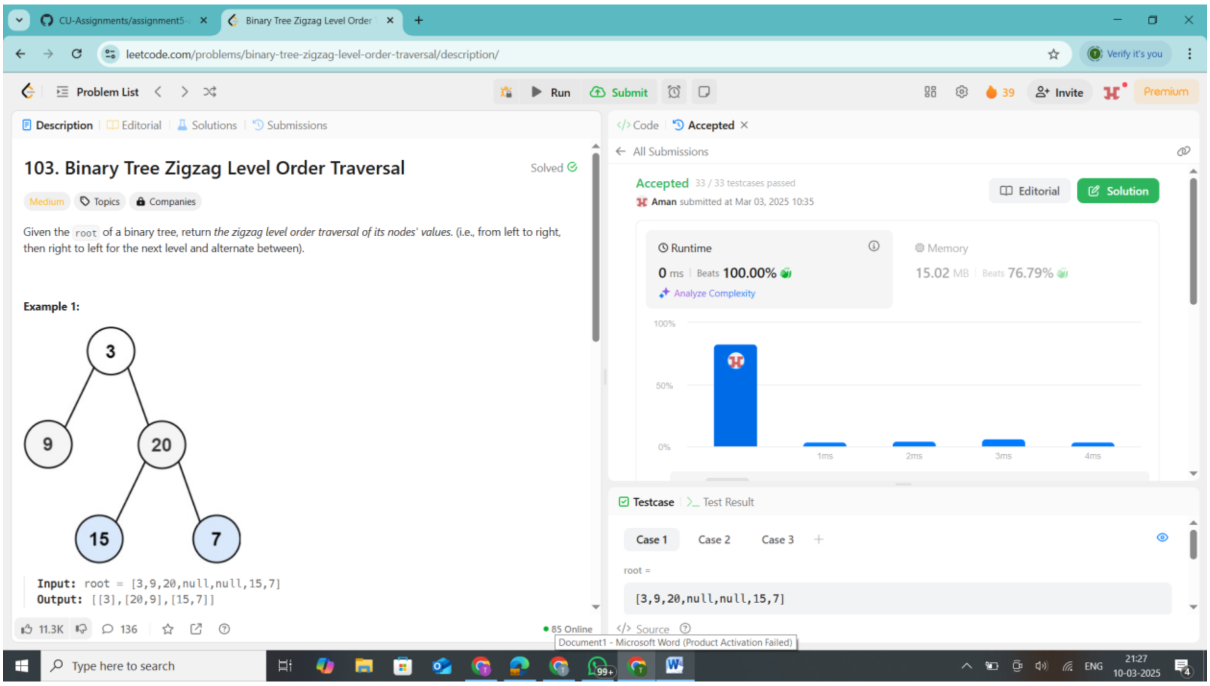
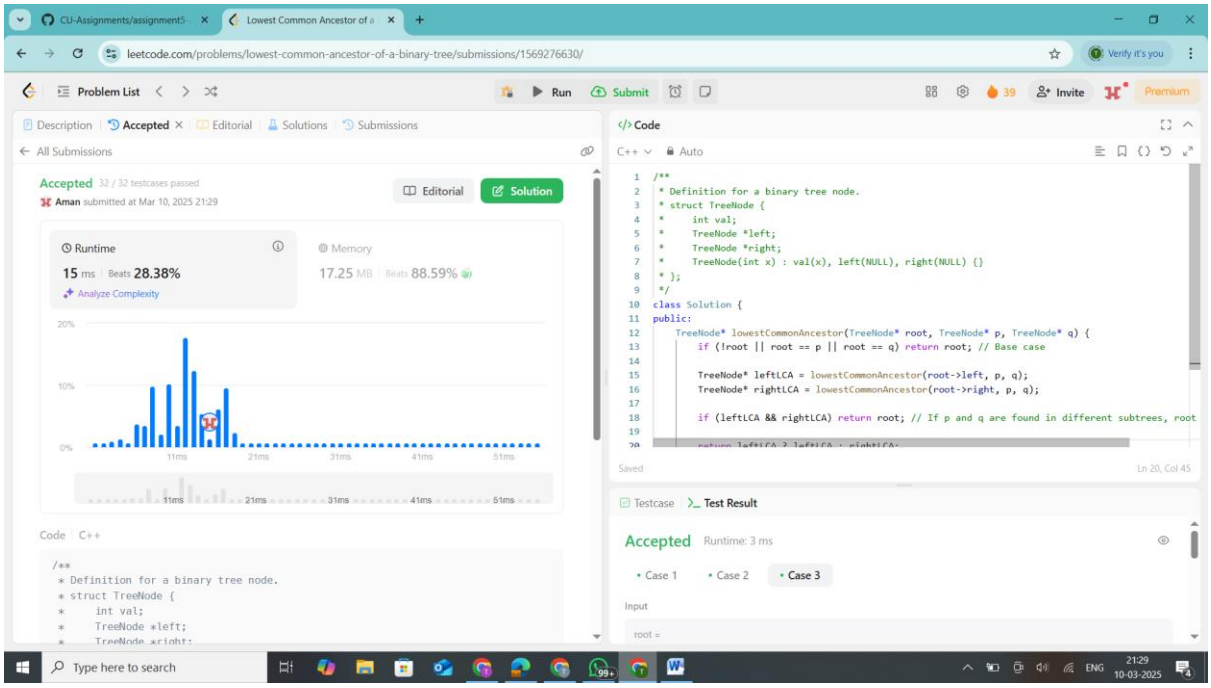
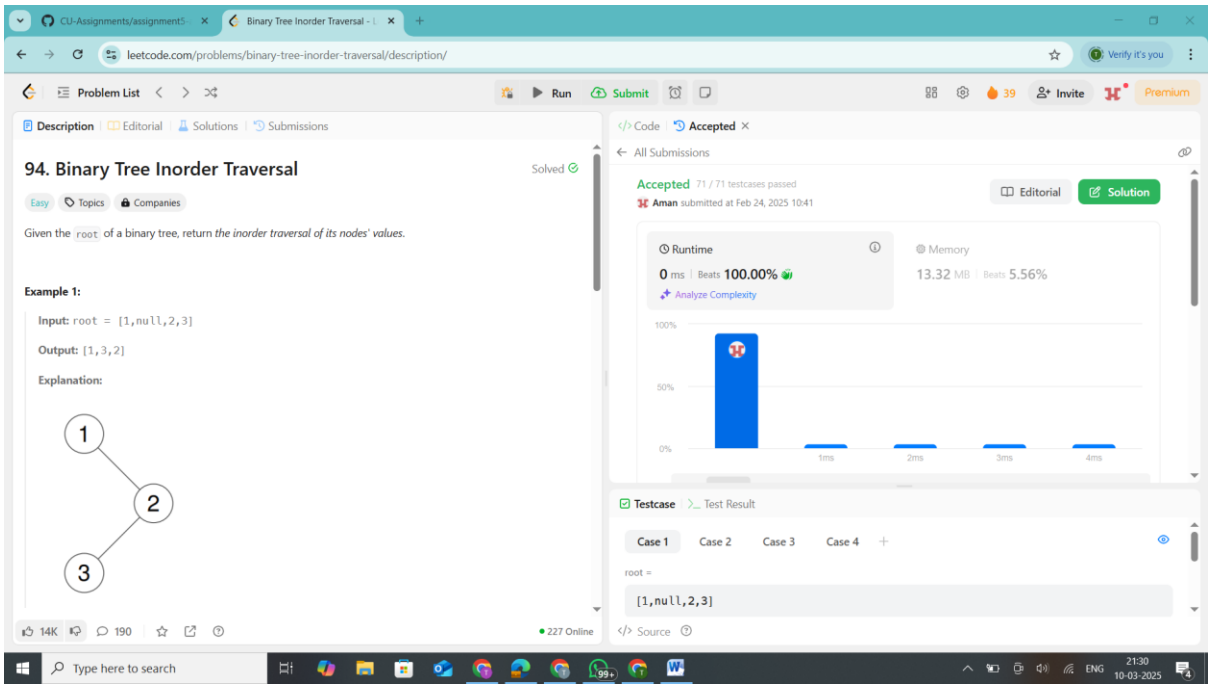

leetcode.com/problems/maximum-depth-of-binary-tree/submissions/1569265808/
Problem List
Run
Submit
Invite
Premium
Description
Editorial
Solutions
Submissions
104. Maximum Depth of Binary Tree
Solved
Easy
Topics
Companies
Given the root of a binary tree, return its maximum depth.
A binary tree's maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.
Example 1:
3
9
20
15
7
Input: root = [3,9,20,null,null,15,7]
Output: 3
13.4K
159
273 Online
Code
Accepted
All Submissions
Accepted 39 / 39 testcases passed
Aman submitted at Mar 10, 2025 21:19
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
18.96 MB | Beats 75.88%
100%
50%
0%
1ms
2ms
3ms
4ms
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
root =
Type here to search
ENG
21:19
10-03-2025

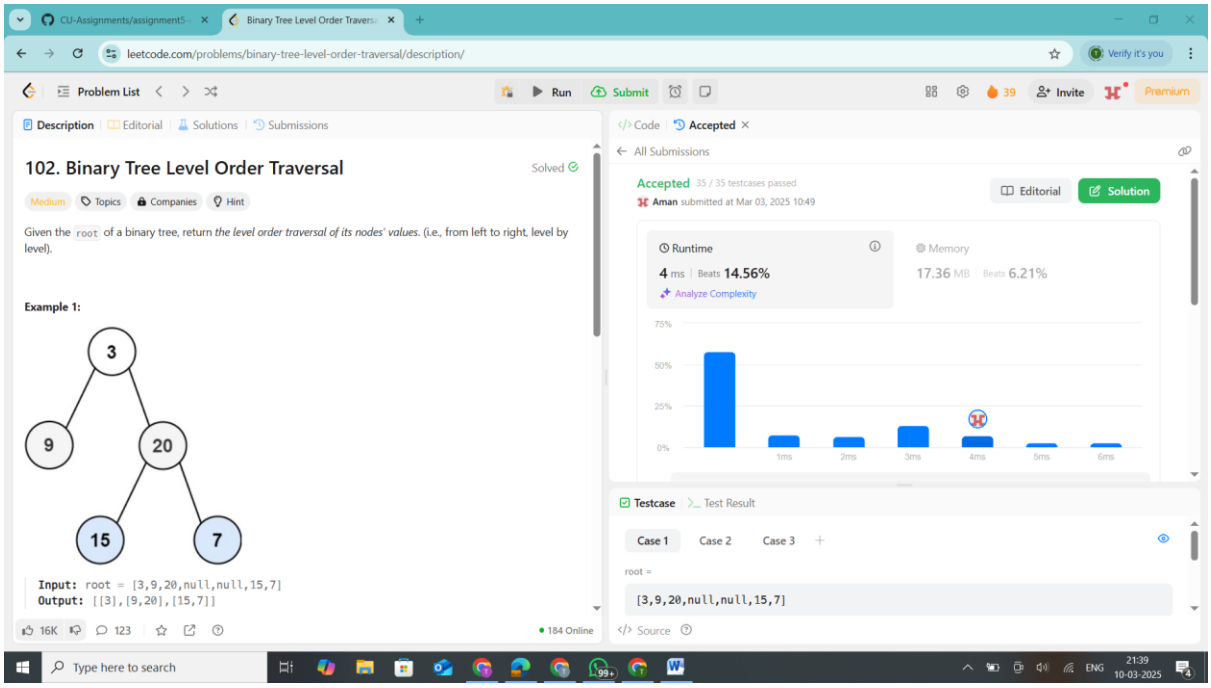

leetcode.com/problems/validate-binary-search-tree/submissions/1569269116/
Problem List
Run
Submit
Invite
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 86 / 86 testcases passed
Aman submitted at Mar 10, 2025 21:22
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
21.91 MB | Beats 48.51%
100%
50%
0%
1ms
2ms
3ms
4ms
Code
C++
Auto
/**
* Definition for a binary tree node.
* struct TreeNode {
*   int val;
*   TreeNode *left;
*   TreeNode *right;
*   TreeNode() : val(0), left(nullptr), right(nullptr) {}
*   TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
*   TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
* };
*/
#include <limits.h>
class Solution {
public:
bool validate(TreeNode* node, long minVal, long maxVal) {
if (!node) return true;
if (node->val <= minVal || node->val >= maxVal) return false;
return validate(node->left, minVal, node->val) && validate(node->right, node->val, maxVal)
}
Saved
Ln 21, Col 51
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
root =
Type here to search
ENG
21:23
10-03-2025

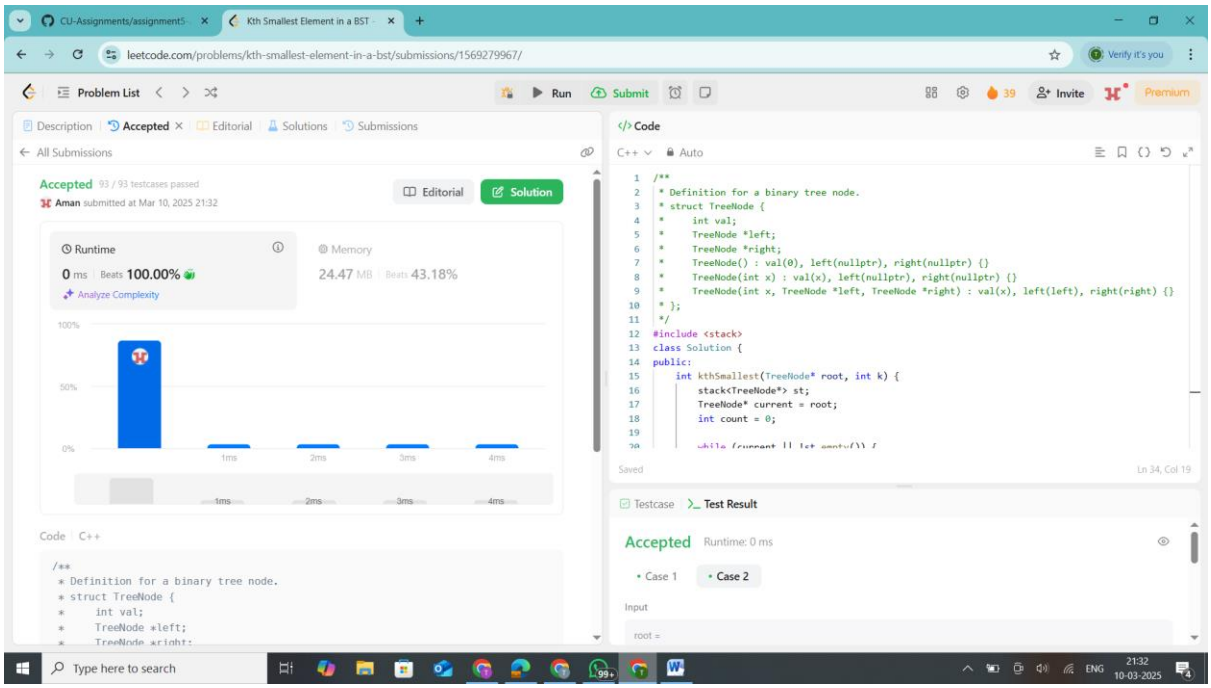

Symmetric Tree - LeetCode
leetcode.com/problems/symmetric-tree/submissions/1569273400/
Accepted 200 / 200 testcases passed
Aman submitted at Mar 10, 2025 21:26
Runtime
0 ms Beats 100.00%
Memory
18.55 MB Beats 27.46%
Code C++
TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
#include <queue>
class Solution {
public:
bool isMirror(TreeNode* t1, TreeNode* t2) {
if (!t1 && !t2) return true; // Both nodes are null
if (!t1 || !t2) return false; // One node is null
return (t1->val == t2->val) && isMirror(t1->left, t2->right) && isMirror(t1->right, t2->le
}
bool isSymmetric(TreeNode* root) {
if (!root) return true;
return isMirror(root->left, root->right);
}
};
Accepted Runtime: 0 ms
Case 1
Case 2
Type here to search
21:26
10-03-2025

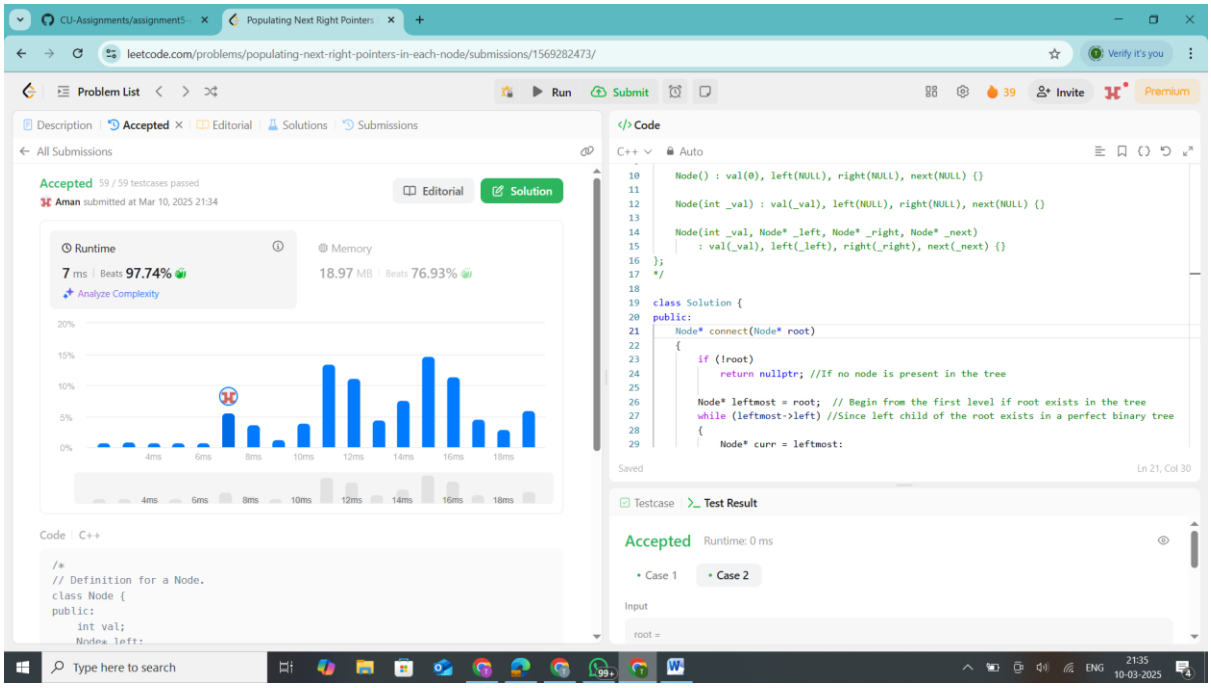

Binary Tree Zigzag Level Order
leetcode.com/problems/binary-tree-zigzag-level-order-traversal/description/
103. Binary Tree Zigzag Level Order Traversal
Solved
Medium
Topics
Companies
Given the root of a binary tree, return the zigzag level order traversal of its nodes' values. (i.e., from left to right, then right to left for the next level and alternate between).
Example 1:
3
9
20
15
7
Input: root = [3,9,20,null,null,15,7]
Output: [[3],[20,9],[15,7]]
Accepted 33 / 33 testcases passed
Aman submitted at Mar 03, 2025 10:35
Runtime
0 ms Beats 100.00%
Memory
15.02 MB Beats 76.79%
Testcase
Case 1
Case 2
Case 3
root =
[3,9,20,null,null,15,7]
Type here to search
21:27
10-03-2025

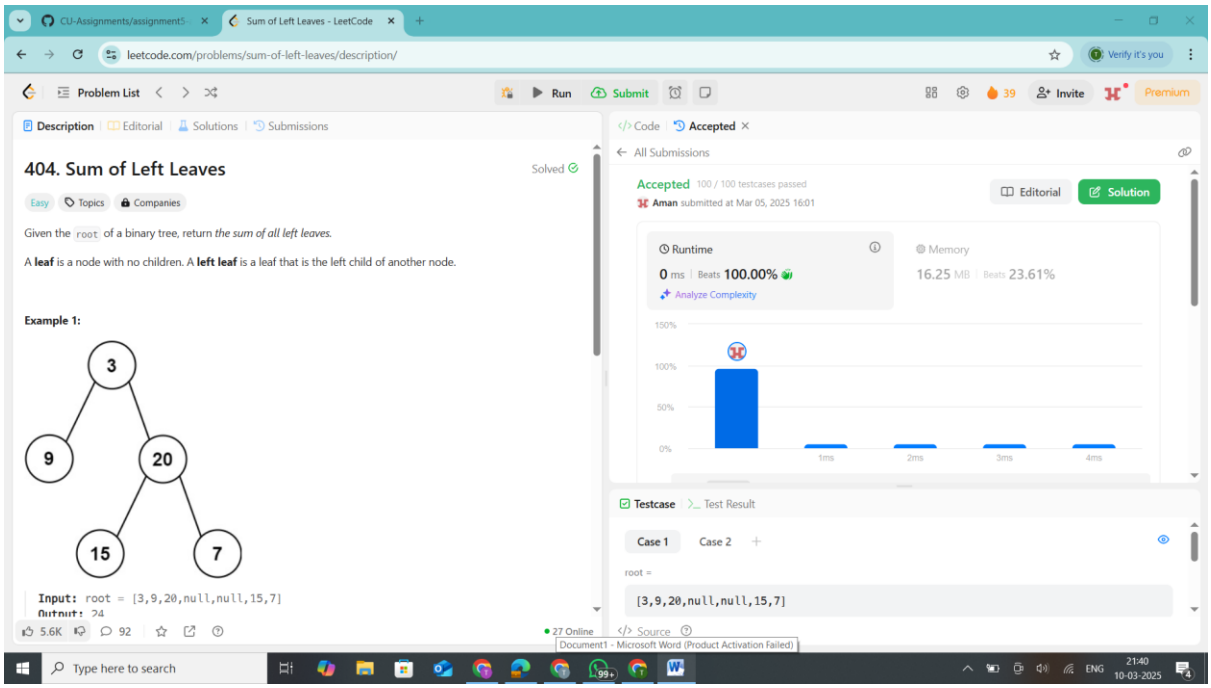

leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree/submissions/1569276630/
Problem List
Run
Submit
Invite
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 32 / 32 testcases passed
Aman submitted at Mar 10, 2025 21:29
Editorial
Solution
Runtime
15 ms | Beats 28.38%
Analyze Complexity
Memory
17.25 MB | Beats 88.59%
Code
C++
Auto
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
class Solution {
public:
TreeNode* lowestCommonAncestor(TreeNode* root, TreeNode* p, TreeNode* q) {
if (!root || root == p || root == q) return root; // Base case
TreeNode* leftLCA = lowestCommonAncestor(root->left, p, q);
TreeNode* rightLCA = lowestCommonAncestor(root->right, p, q);
if (leftLCA && rightLCA) return root; // If p and q are found in different subtrees, root
Saved
Ln 20, Col 45
Testcase
Test Result
Accepted
Runtime: 3 ms
Case 1
Case 2
Case 3
Input
root =
Type here to search
ENG
21:29
10-03-2025


leetcode.com/problems/binary-tree-inorder-traversal/description/
Problem List
Run
Submit
Invite
Premium
Description
Editorial
Solutions
Submissions
94. Binary Tree Inorder Traversal
Solved
Easy
Topics
Companies
Given the root of a binary tree, return the inorder traversal of its nodes' values.
Example 1:
Input: root = [1,null,2,3]
Output: [1,3,2]
Explanation:
1
2
3
14K
190
227 Online
Code
Accepted
All Submissions
Accepted 71 / 71 testcases passed
Aman submitted at Feb 24, 2025 10:41
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
13.32 MB | Beats 5.56%
Testcase
Test Result
Case 1
Case 2
Case 3
Case 4
root =
[1,null,2,3]
Source
Type here to search
ENG
21:30
10-03-2025


leetcode.com/problems/binary-tree-level-order-traversal/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
102. Binary Tree Level Order Traversal
Solved
Medium
Topics
Companies
Hint
Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).
Example 1:
3
9
20
15
7
Input: root = [3,9,20,null,null,15,7]
Output: [[3],[9,20],[15,7]]
16K
123
184 Online
Code
Accepted
All Submissions
Accepted 35 / 35 testcases passed
Aman submitted at Mar 03, 2025 10:49
Editorial
Solution
Runtime
4 ms | Beats 14.56%
Analyze Complexity
Memory
17.36 MB | Beats 6.21%
75%
50%
25%
0%
1ms
2ms
3ms
4ms
5ms
6ms
Testcase
Test Result
Case 1
Case 2
Case 3
root =
[3,9,20,null,null,15,7]
Source
Type here to search
ENG
21:39
10-03-2025


leetcode.com/problems/kth-smallest-element-in-a-bst/submissions/1569279967/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 93 / 93 testcases passed
Aman submitted at Mar 10, 2025 21:32
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
24.47 MB | Beats 43.18%
100%
50%
0%
1ms
2ms
3ms
4ms
Code
C++
Auto
/**
* Definition for a binary tree node.
* struct TreeNode {
*   int val;
*   TreeNode *left;
*   TreeNode *right;
*   TreeNode() : val(0), left(nullptr), right(nullptr) {}
*   TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
*   TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
* };
*/
#include <stack>
class Solution {
public:
int kthSmallest(TreeNode* root, int k) {
stack<TreeNode*> st;
TreeNode* current = root;
int count = 0;
Saved
Ln 34, Col 19
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
root =
Type here to search
ENG
21:32
10-03-2025


Populating Next Right Pointers
leetcode.com/problems/populating-next-right-pointers-in-each-node/submissions/1569282473/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 59 / 59 testcases passed
Aman submitted at Mar 10, 2025 21:34
Editorial
Solution
Runtime
7 ms | Beats 97.74%
Analyze Complexity
Memory
18.97 MB | Beats 76.93%
Code
C++
Auto
Node() : val(0), left(NULL), right(NULL), next(NULL) {}
Node(int _val) : val(_val), left(NULL), right(NULL), next(NULL) {}
Node(int _val, Node* _left, Node* _right, Node* _next)
: val(_val), left(_left), right(_right), next(_next) {}
};
*/
class Solution {
public:
Node* connect(Node* root)
{
if (!root)
return nullptr; //If no node is present in the tree
Node* leftmost = root; // Begin from the first level if root exists in the tree
while (leftmost->left) //Since left child of the root exists in a perfect binary tree
{
Node* curr = leftmost:
Saved
Ln 21, Col 30
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
root =
/*
// Definition for a Node.
class Node {
public:
int val;
Type here to search
ENG
21:35
10-03-2025


Sum of Left Leaves - LeetCode
leetcode.com/problems/sum-of-left-leaves/description/
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
404. Sum of Left Leaves
Solved
Easy
Topics
Companies
Given the root of a binary tree, return the sum of all left leaves.
A leaf is a node with no children. A left leaf is a leaf that is the left child of another node.
Example 1:
3
9
20
15
7
Input: root = [3,9,20,null,null,15,7]
Code
Accepted
All Submissions
Accepted 100 / 100 testcases passed
Aman submitted at Mar 05, 2025 16:01
Editorial
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
16.25 MB | Beats 23.61%
Testcase
Test Result
Case 1
Case 2
root =
[3,9,20,null,null,15,7]
5.6K
92
27 Online
Source
Document1 - Microsoft Word (Product Activation Failed)
Type here to search
ENG
21:40
10-03-2025