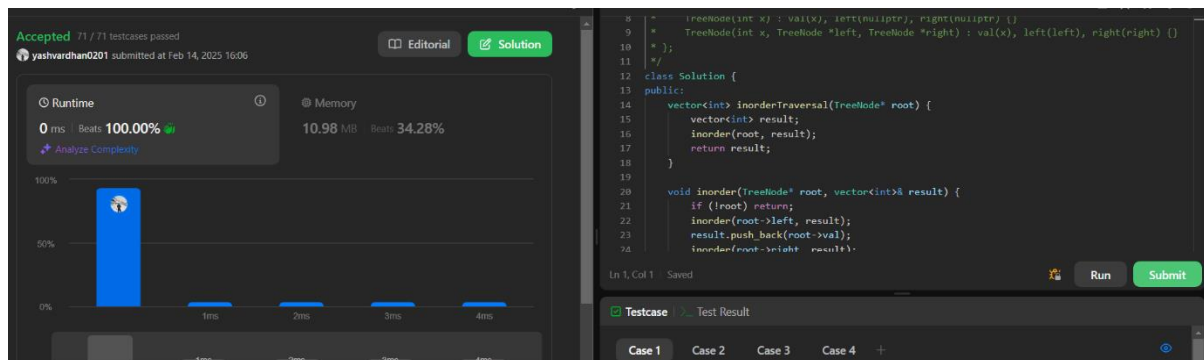# ASSIGNMENT-3(AP)

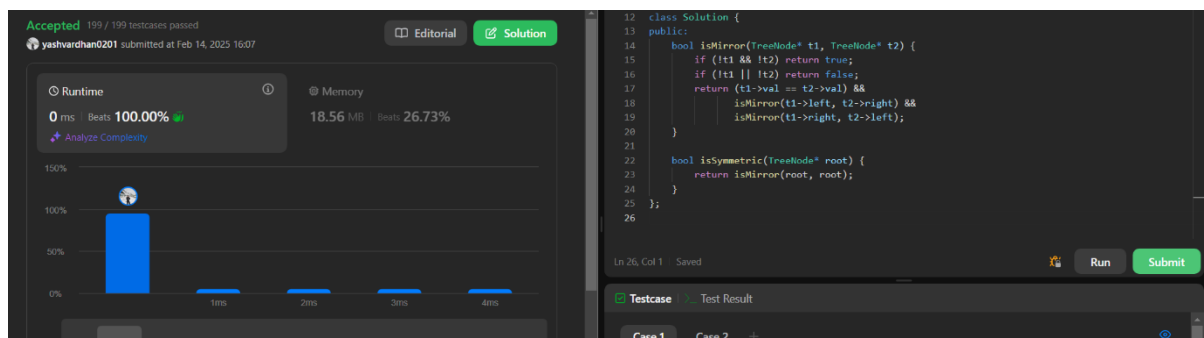**Name:** Yashvardhan Singh

UID: 22BCS10258
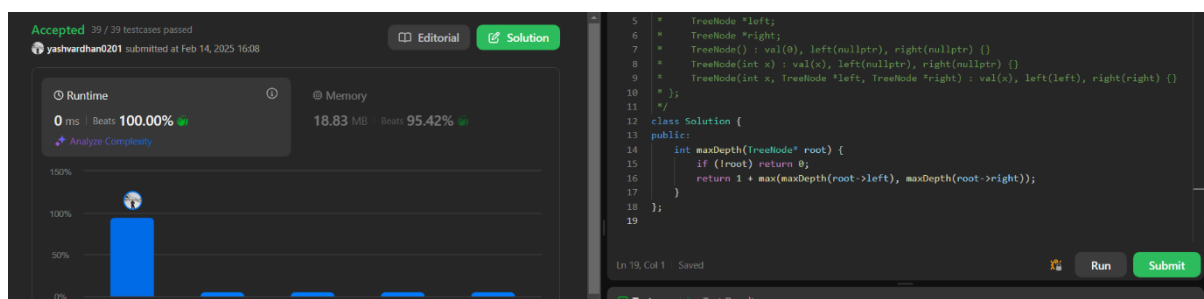
Section: 22BCS_FL_IOT-604

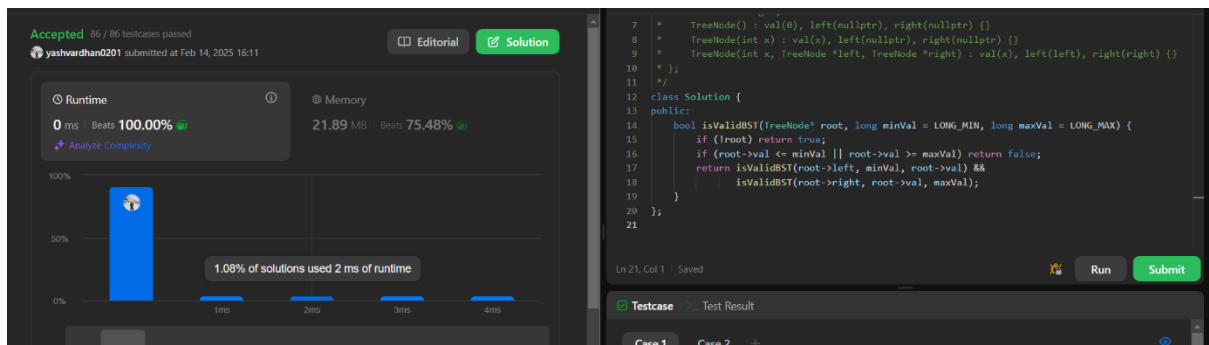**Group:** A

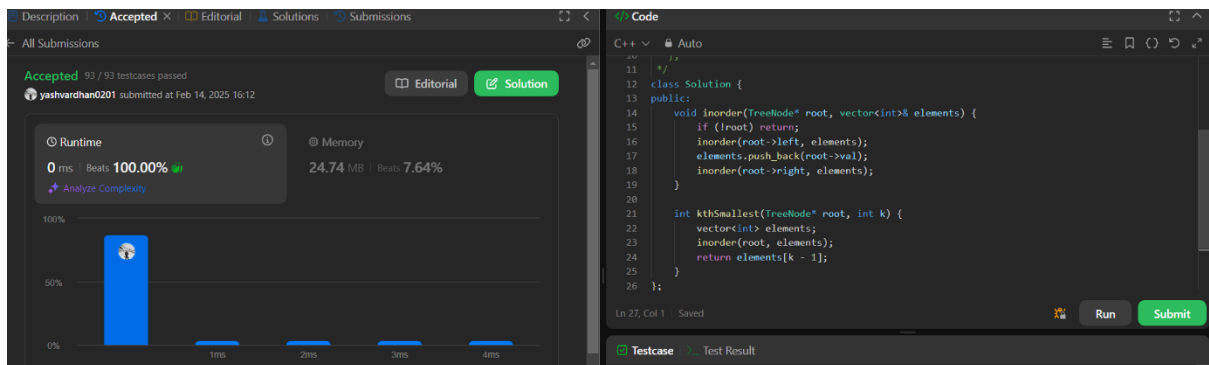## 1) binary-tree-inorder-traversal

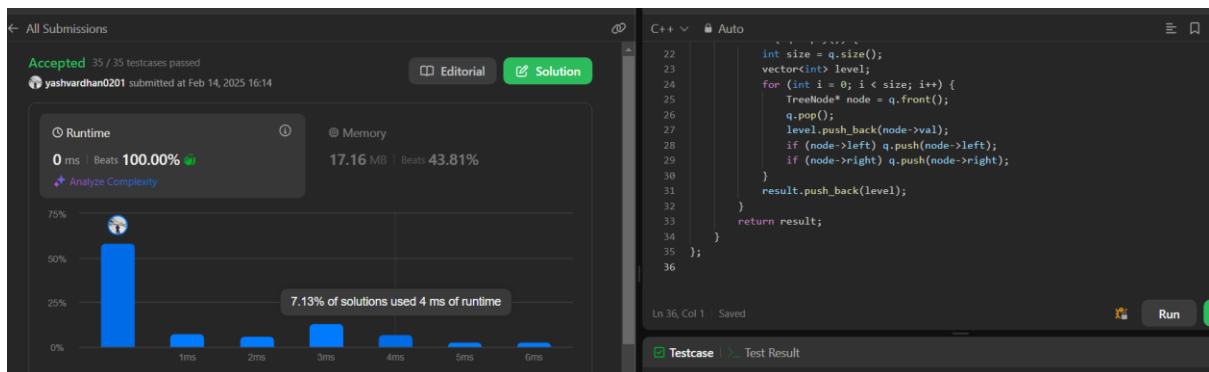
2) symmetric-tree

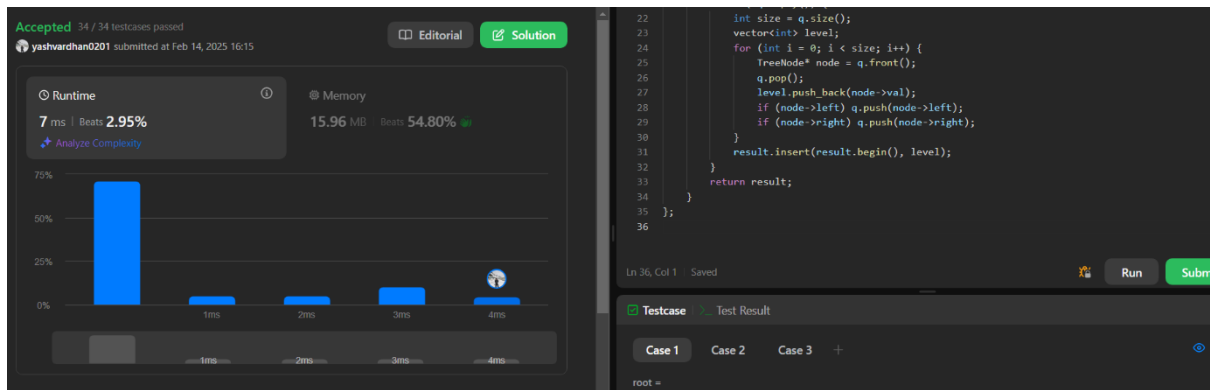## 3) maximum-depth-of-binary-tree

4) validate-binary-search-tree

Accepted 86 / 86 testcases passed
yashvardhan0201 submitted at Feb 14, 2025 16:11

Runtime
0 ms | Beats **100.00%**

Memory
21.89 MB | Beats **75.48%**

1.08% of solutions used 2 ms of runtime

```cpp
 7  *      TreeNode() : val(0), left(nullptr), right(nullptr) {}
 8  *      TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 9  *      TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
10  * };
11  */
12  class Solution {
13  public:
14      bool isValidBST(TreeNode* root, long minVal = LONG_MIN, long maxVal = LONG_MAX) {
15          if (!root) return true;
16          if (root->val <= minVal || root->val >= maxVal) return false;
17          return isValidBST(root->left, minVal, root->val) &&
18                 isValidBST(root->right, root->val, maxVal);
19      }
20  };
21
```

5) kth-smallest-element-in-a-bst

Accepted 93 / 93 testcases passed
yashvardhan0201 submitted at Feb 14, 2025 16:12

Runtime
0 ms | Beats **100.00%**

Memory
24.74 MB | Beats **7.64%**

```cpp
11  */
12  class Solution {
13  public:
14      void inorder(TreeNode* root, vector<int>& elements) {
15          if (!root) return;
16          inorder(root->left, elements);
17          elements.push_back(root->val);
18          inorder(root->right, elements);
19      }
20
21      int kthSmallest(TreeNode* root, int k) {
22          vector<int> elements;
23          inorder(root, elements);
24          return elements[k - 1];
25      }
26  };
```

6) binary-tree-level-order-traversal

Accepted 35 / 35 testcases passed
yashvardhan0201 submitted at Feb 14, 2025 16:14

Runtime
0 ms | Beats **100.00%**

Memory
17.16 MB | Beats **43.81%**

7.13% of solutions used 4 ms of runtime

```cpp
22              int size = q.size();
23              vector<int> level;
24              for (int i = 0; i < size; i++) {
25                  TreeNode* node = q.front();
26                  q.pop();
27                  level.push_back(node->val);
28                  if (node->left) q.push(node->left);
29                  if (node->right) q.push(node->right);
30              }
31              result.push_back(level);
32          }
33          return result;
34      }
35  };
36
```
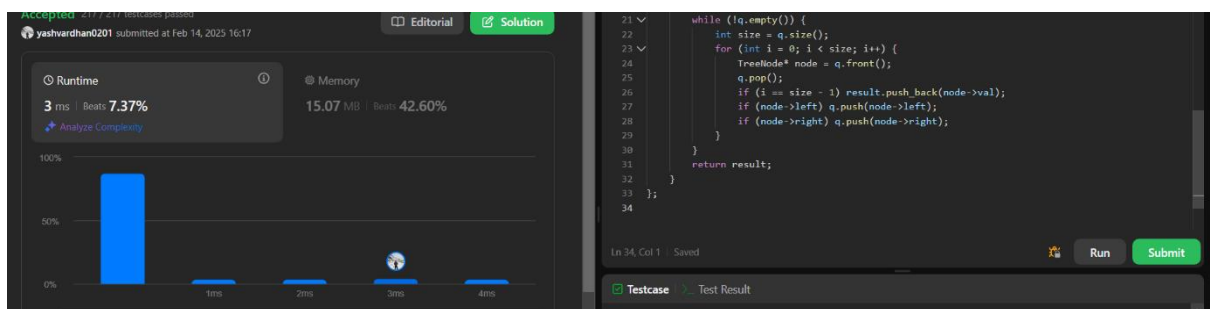
7) Binary Tree Level Order Traversal II

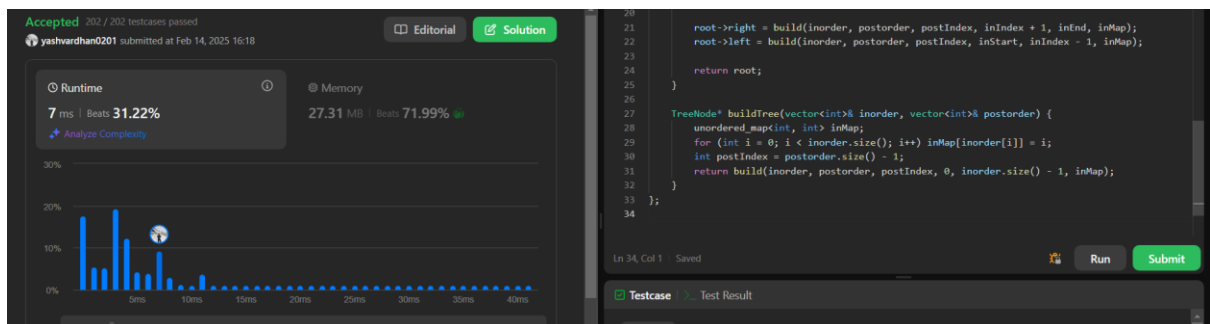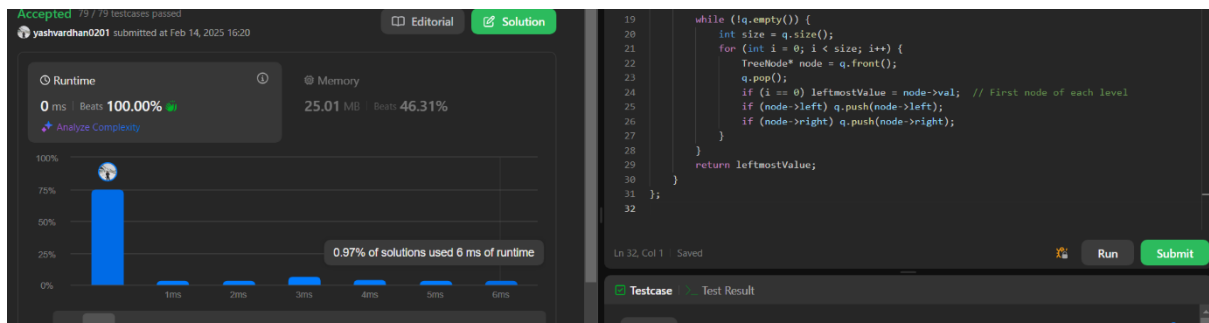## 8) Binary Tree Zigzag Level Order Traversal
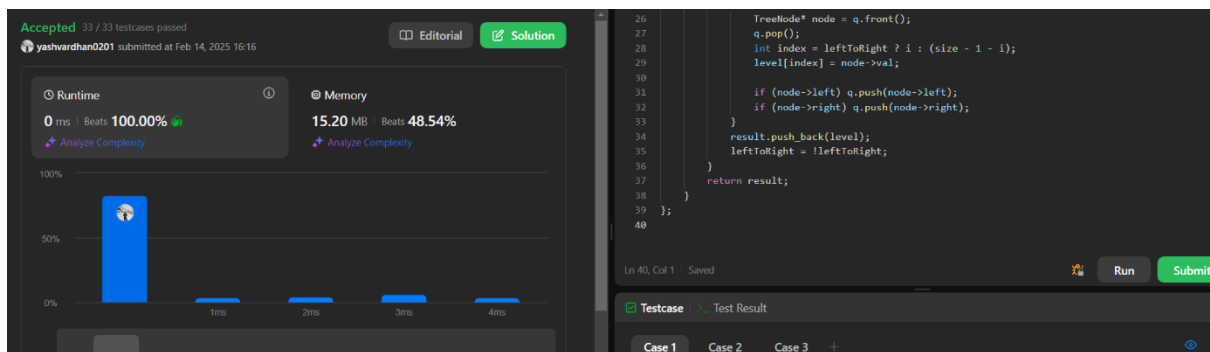
9) Binary Tree Right Side View

10) .Construct Binary Tree from Inorder and Postorder Traversal

11) Find Bottom Left Tree Value

```
19      while (!q.empty()) {
20          int size = q.size();
21          for (int i = 0; i < size; i++) {
22              TreeNode* node = q.front();
23              q.pop();
24              if (i == 0) leftmostValue = node->val;  // First node of each level
25              if (node->left) q.push(node->left);
26              if (node->right) q.push(node->right);
27          }
28      }
29      return leftmostValue;
30  }
31  };
32
```

12) Binary Tree Maximum Path Sum


```
18
19          int left = max(0, maxPath(root->left))
20          int right = max(0, maxPath(root->right
21
22          maxSum = max(maxSum, left + right + ro
23
24          return root->val + max(left, right);
25      }
26
27      int maxPathSum(TreeNode* root) {
28          maxPath(root);
29          return maxSum;
30      }
31  };
32
```
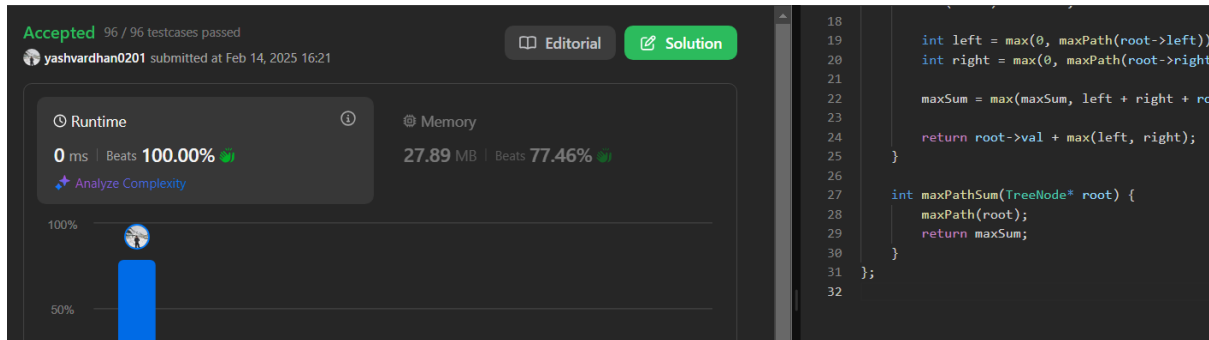
13) Vertical Order Traversal of a Binary Tree


```
8   *    TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
9   *    TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left)
10  * };
11  */
12  class Solution {
13  public:
14      vector<vector<int>> verticalTraversal(TreeNode* root) {
15          map<int, map<int, multiset<int>>> nodes;
16          queue<pair<TreeNode*, pair<int, int>>> q;
17          q.push({root, {0, 0}});
18
19          while (!q.empty()) {
20              auto [node, pos] = q.front();
21              q.pop();
22              int x = pos.first, y = pos.second;
23
```