Experiment 5

Student Name: Shivam Kumar
UID: 22BCS10524
Branch: BE-CSE
Section/Group: 22BCS_IOT-641/A
Semester: 6th
Subject Code: 22CSP-351
Subject Name: AP Lab-II

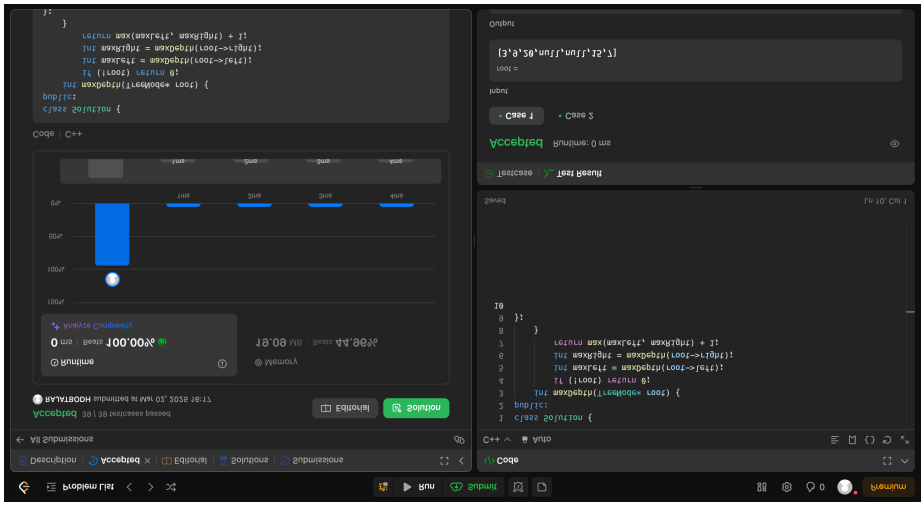
A. Maximum Depth of Binary Tree

1. Aim: Given the root of a binary tree, return its maximum depth.

2. Code

```
class Solution {
public:
    int maxDepth(TreeNode* root) {
        if (!root) return 0;
        int maxLeft = maxDepth(root->left);
        int maxRight = maxDepth(root->right);
        return max(maxLeft, maxRight) + 1;
    }
};
```

3. Output:



4. Link: https://leetcode.com/problems/maximum-depth-of-binary-tree/submissions/1560269754/

B. Validate Binary Search Tree

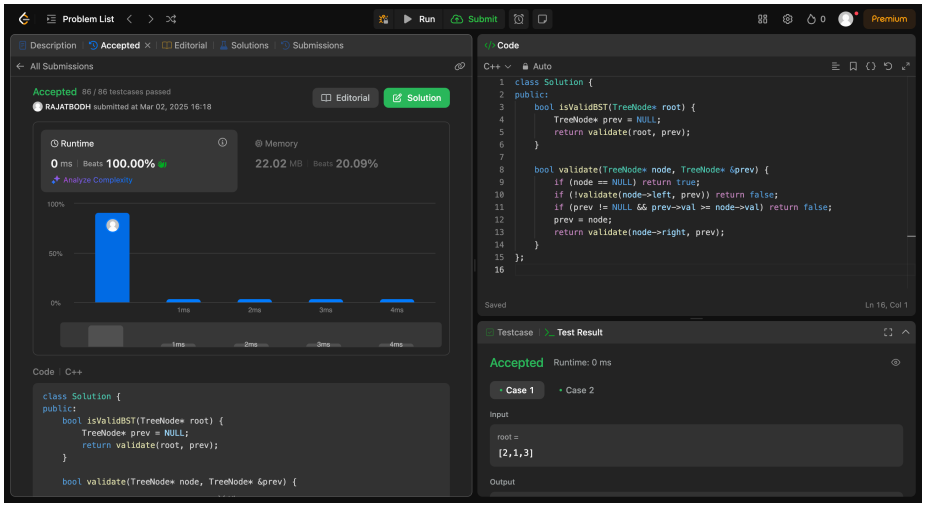
1. Aim: Given the root of a binary tree, determine if it is a valid binary search tree (BST).

2. Code

```
class Solution {
public:
    bool isValidBST(TreeNode* root) {
        TreeNode* prev = NULL;
        return validate(root, prev);
    }

    bool validate(TreeNode* node, TreeNode* &prev) {
        if (node == NULL) return true;
        if (!validate(node->left, prev)) return false;
        if (prev != NULL && prev->val >= node->val) return false;
        prev = node;
        return validate(node->right, prev);
    }
};
```

3. Output:



4. Link: https://leetcode.com/problems/validate-binary-search-tree/submissions/1560270579/

C. Symmetric Tree

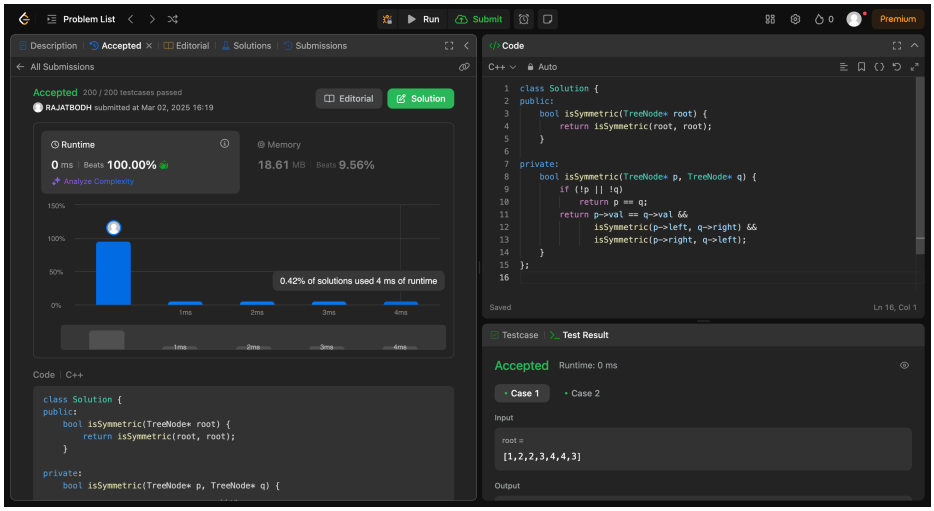
1. **Aim:** Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

2. Code

```
class Solution {
public:
    bool isSymmetric(TreeNode* root) {
        return isSymmetric(root, root);
    }

private:
    bool isSymmetric(TreeNode* p, TreeNode* q) {
        if (!p || !q)
            return p == q;
        return p->val == q->val &&
               isSymmetric(p->left, q->right) &&
               isSymmetric(p->right, q->left);
    }
};
```

3. Output:



4. **Link:** https://leetcode.com/problems/symmetric-tree/submissions/1560271585/