Experiment-5

Student Name: Jay Ananya **UID: 22BCS10822**

Branch: BE-CSE **Section/Group: KPIT_901 (B)**

Semester: 6th **Date of Performance: 27/02/2025**

Subject Name: AP LAB-II **Subject Code: 22CSP-351**

Problem- 1

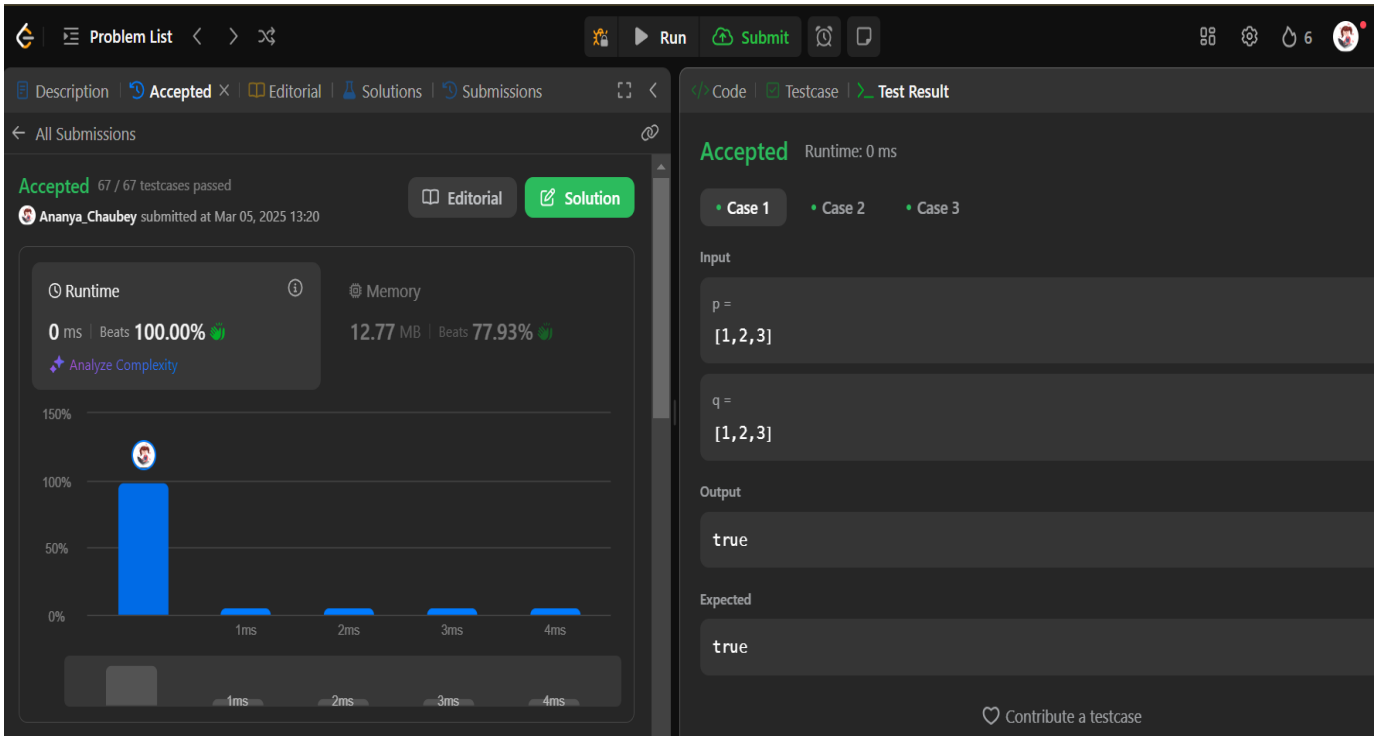
1. Aim:

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.

2. Implementation/Code: Backend:

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

3. Output:



4. Learning Outcomes:

a. Understanding binary tree structure
b. Implementing recursive tree traversal
c. Comparing two trees for identical structure and values
i. Handling edge cases like empty trees

Problem- 2

1. **Aim:**

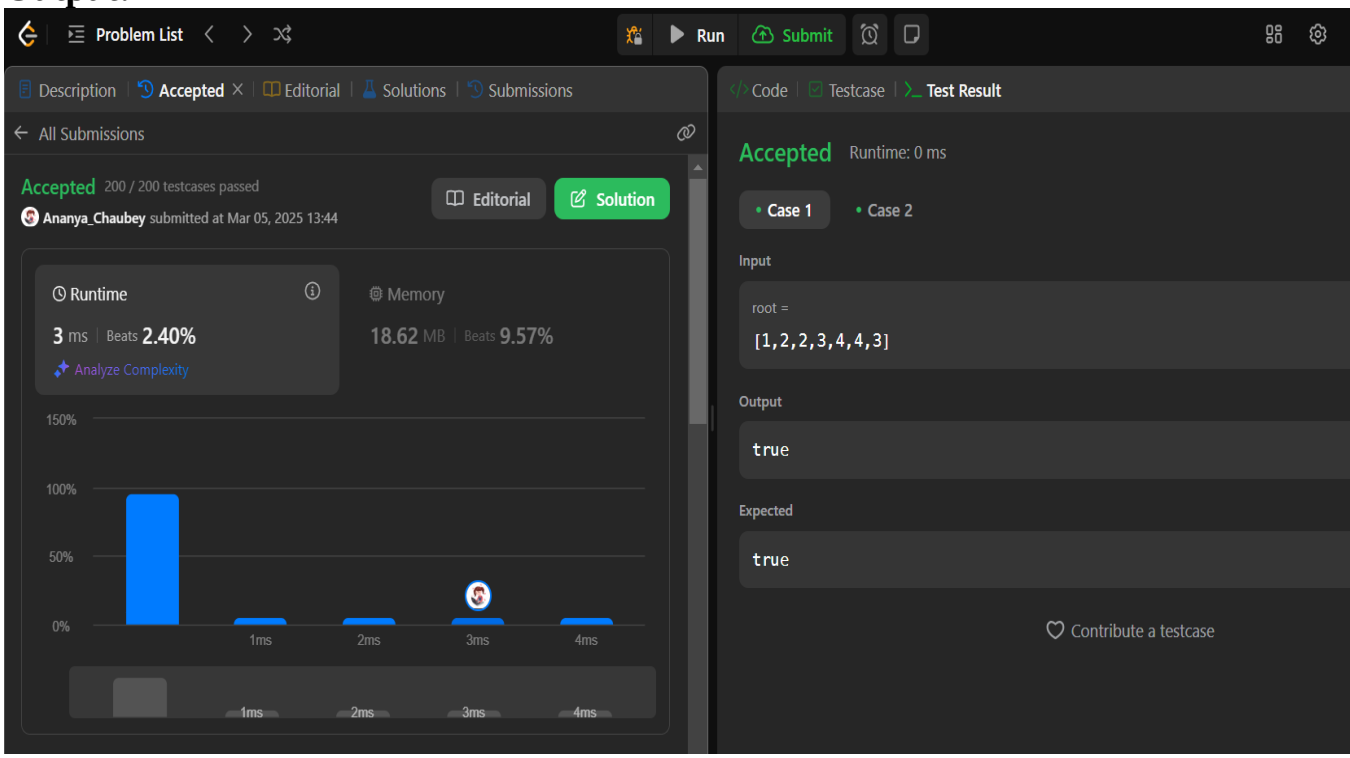
   Given the root of a binary tree, check whether it is amirror of itself (i.e., symmetric around its center).

2. **Implementation/Code: Backend:**

```
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. Learning Outcomes:

- Understanding binary tree structure
- Implementing recursion for tree traversal
- Checking symmetry using mirror property
- Handling edge cases like empty trees