Experiment-5

Student Name: Siddharth Lal **UID: 22BCS13410**

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

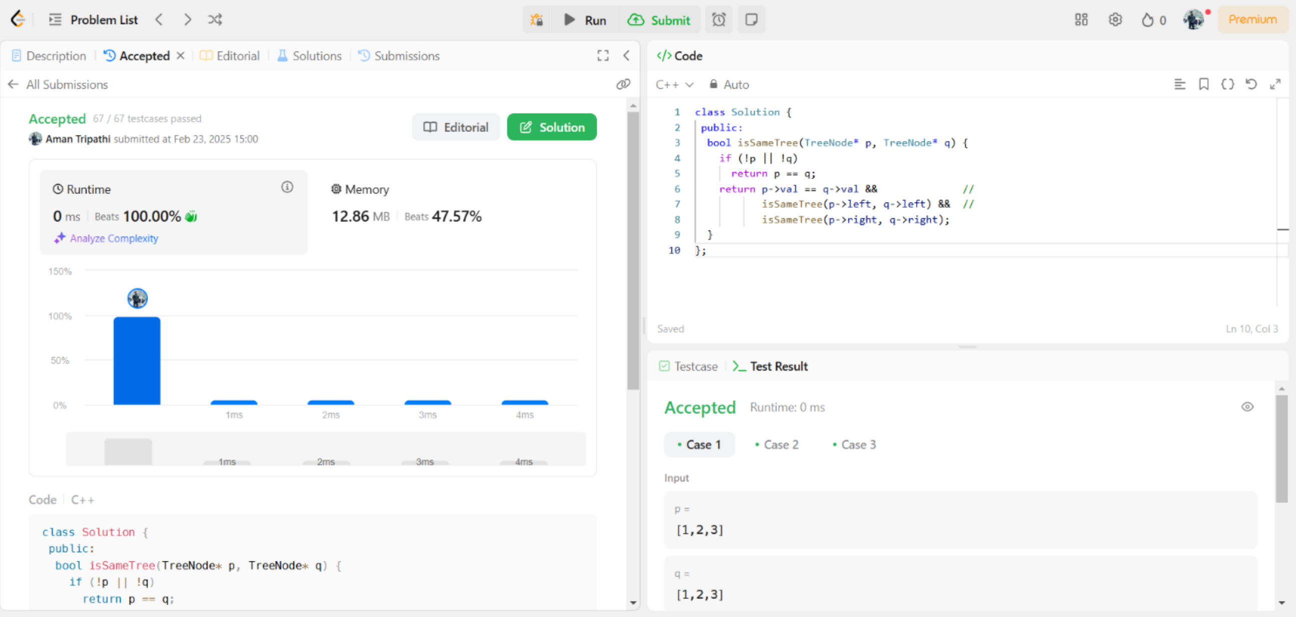
1. Aim:

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:



Accepted 67 / 67 testcases passed

Aman Tripathi submitted at Feb 23, 2025 15:00

Runtime
0 ms | Beats 100.00%

Memory
12.86 MB | Beats 47.57%

Code | C++

```cpp
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);
  }
};
```

Accepted Runtime: 0 ms

Input

p =
[1,2,3]

q =
[1,2,3]

4. Learning Outcomes:

- Understanding binary tree structure
- Implementing recursive tree traversal
- Comparing two trees for identical structure and values
- Handling edge cases like empty trees

Problem- 2

1. Aim:

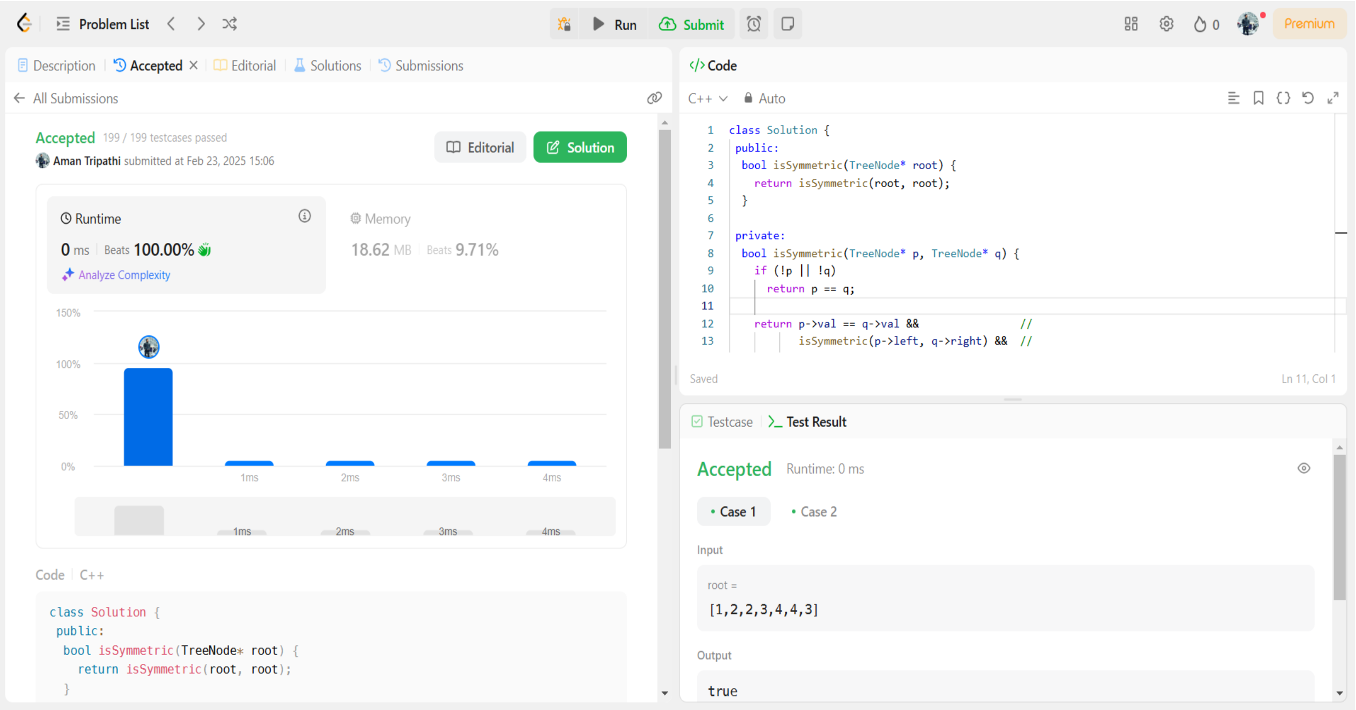
Given the root of a binary tree, check whether it is amirror of itself (i.e., symmetric around its center).

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:



Accepted 199 / 199 testcases passed

Aman Tripathi submitted at Feb 23, 2025 15:06

Runtime
0 ms | Beats 100.00%

Memory
18.62 MB | Beats 9.71%

Code | C++

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

```
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) &&  //
```

Saved

Ln 11, Col 1

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

root =

```
[1,2,2,3,4,4,3]
```

Output

```
true
```

4. Learning Outcomes:

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