



## Experiment-5

**Student Name:** Sahil

**UID:** 22BCS10928

**Branch:** BE-CSE

**Section/Group:** KPIT\_901 (B)

**Semester:** 6<sup>th</sup>

**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:

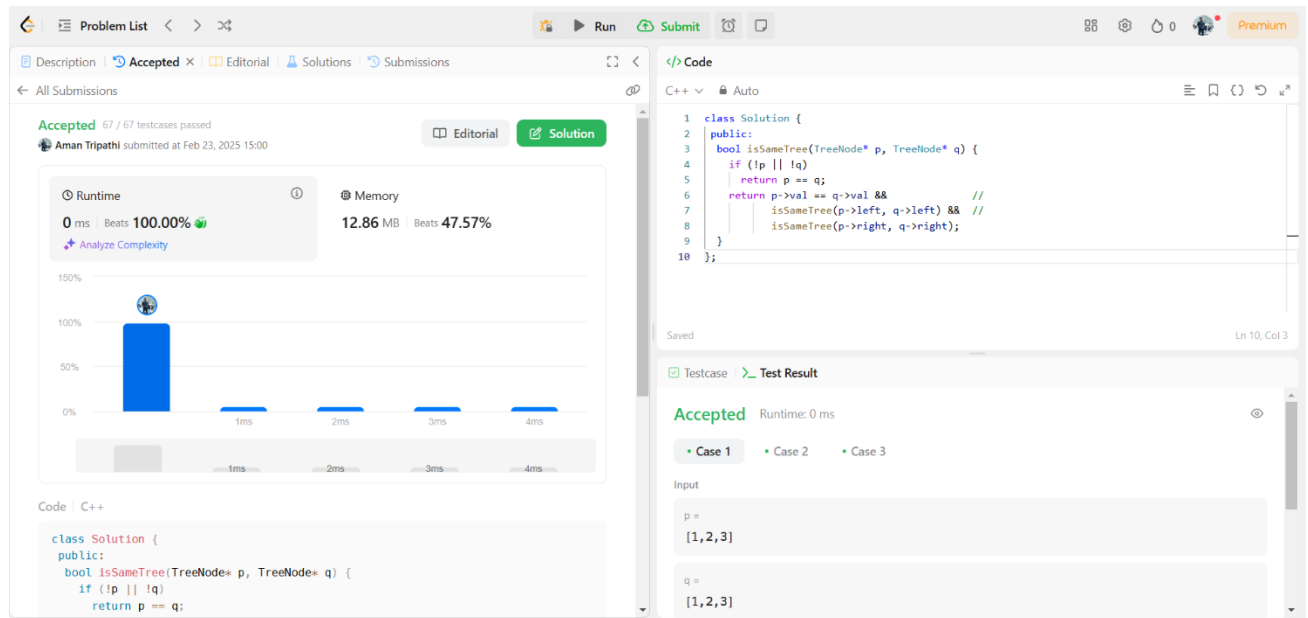
```
class Solution {  
public:  
    bool isSameTree(TreeNode* p, TreeNode* q) {  
        if (!p || !q)  
            return p == q;  
        return p->val == q->val &&  
            isSameTree(p->left, q->left) &&  
            isSameTree(p->right, q->right);  
    }  
};
```

#### 3. Output:



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.



## 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:

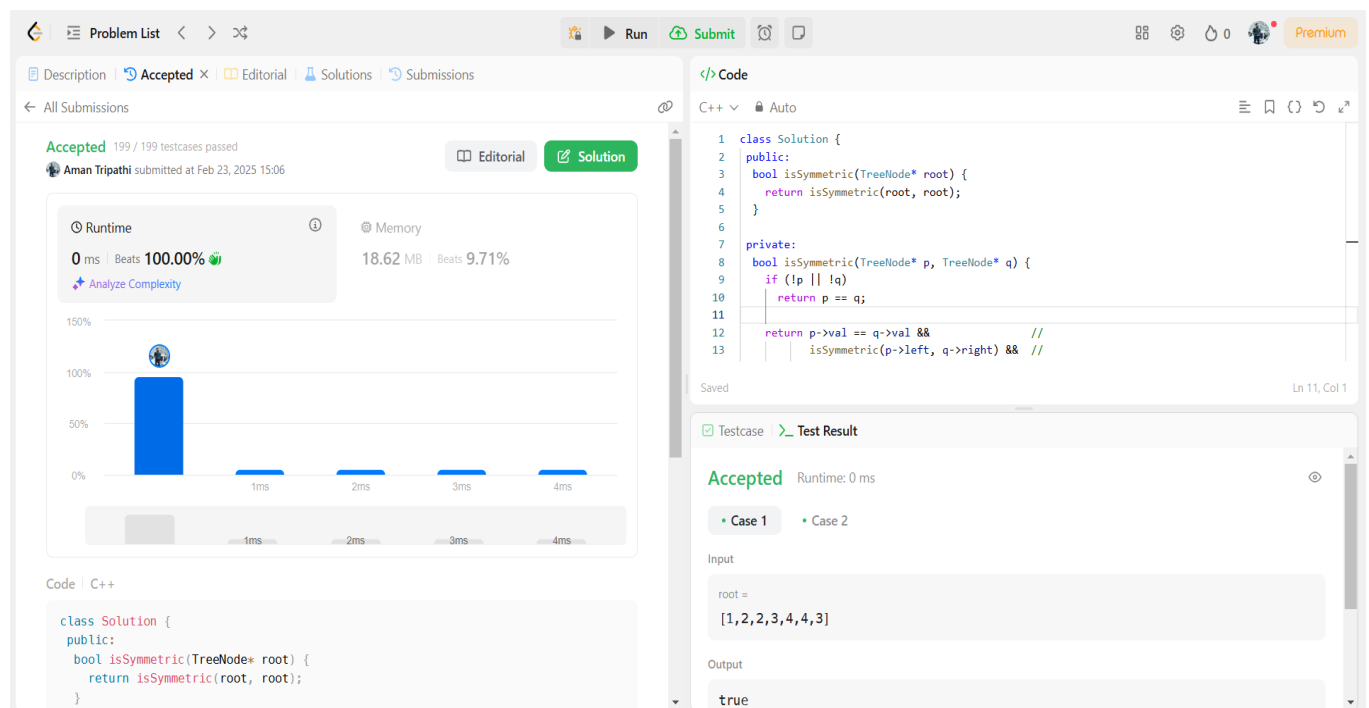
Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

### 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:



The screenshot displays a coding platform interface. On the left, the 'Problem List' shows 'Accepted' status for 199/199 testcases passed. The user 'Aman Tripathi' submitted the solution on Feb 23, 2025, at 15:06. The performance graph shows a runtime of 0 ms, beating 100.00% of other solutions, and a memory usage of 18.62 MB, beating 9.71% of other solutions. The code editor on the right shows the C++ implementation of the `isSymmetric` function. The test result panel on the bottom right shows the input array [1, 2, 2, 3, 4, 4, 3] and the output 'true'.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## 4. Learning Outcomes:

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