



## Experiment-5

**Student Name:** DEVANSH TYAGI

**UID:** 22BCS10874

**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:

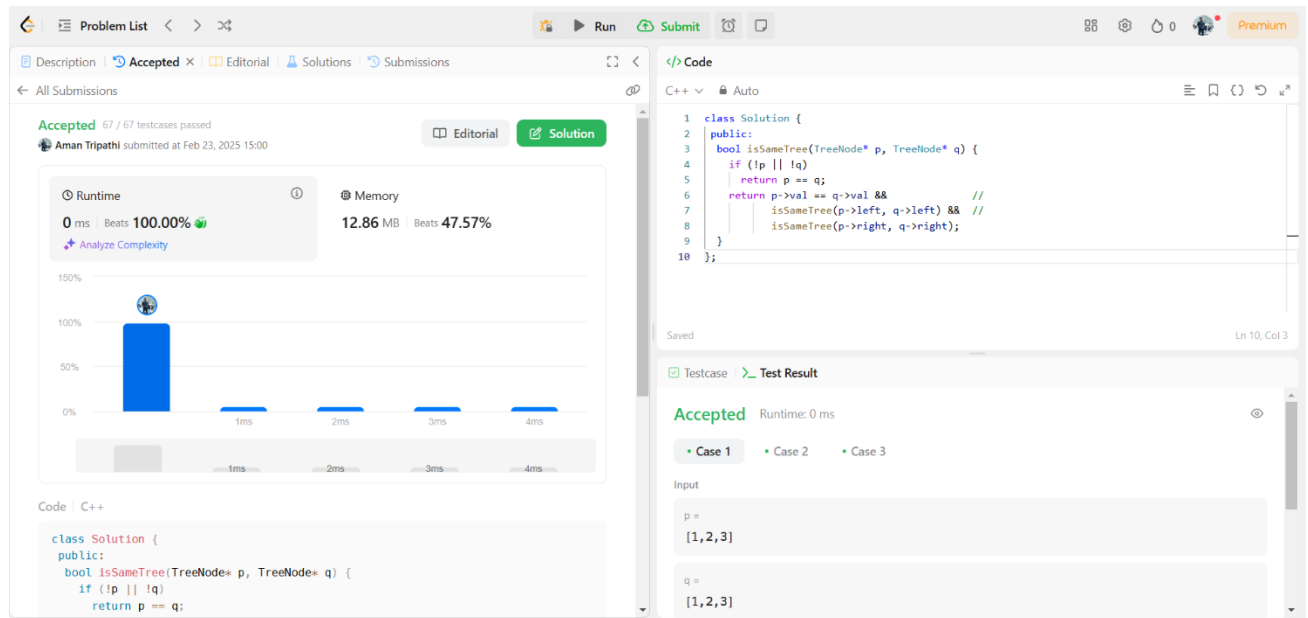
```
class Solution {  
public:  
    bool isSameTree(TreeNode* p, TreeNode* q) {  
        if (!p || !q)  
            return p == q;  
        return p->val == q->val &&  
            isSameTree(p->left, q->left) &&  
            isSameTree(p->right, q->right);  
    }  
};
```

#### 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:

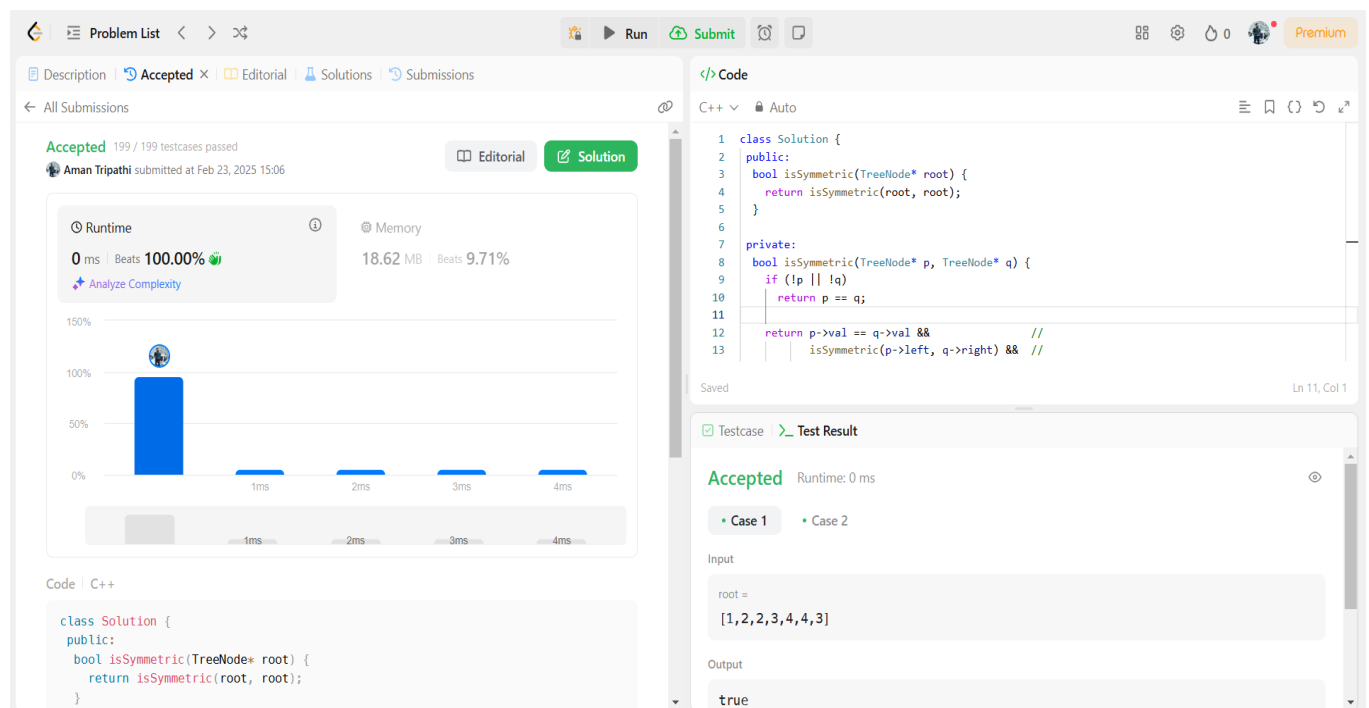
Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

### 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 for a problem titled "Problem- 2". The interface is divided into several sections:

- Problem List:** Shows the current problem selected.
- Description:** Contains the problem statement and the code editor.
- Accepted:** Indicates that the solution is accepted, with 199/199 testcases passed.
- Runtime/Memory:** Shows the execution time (0 ms) and memory usage (18.62 MB), both of which are optimal (Beats 100.00% and 9.71% respectively).
- Code:** Displays the C++ code for the solution, which is a recursive function to check if a binary tree is a mirror of itself.
- Test Result:** Shows the test case results, including the input and output for a specific case.

The input for the test case is:

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

The output is:

```
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