Experiment 1.5

Student Name: Avinash Kumar **UID: 21BCS8908**
Branch: CSE **Section/Group: 648-B**
Semester: 6th **Date of Performance: 27.02.24**
Subject Name: AP-2 **Subject Code: 21CSP-351**

Aim: To demonstrate the concept of Binary Tree.

Objective:

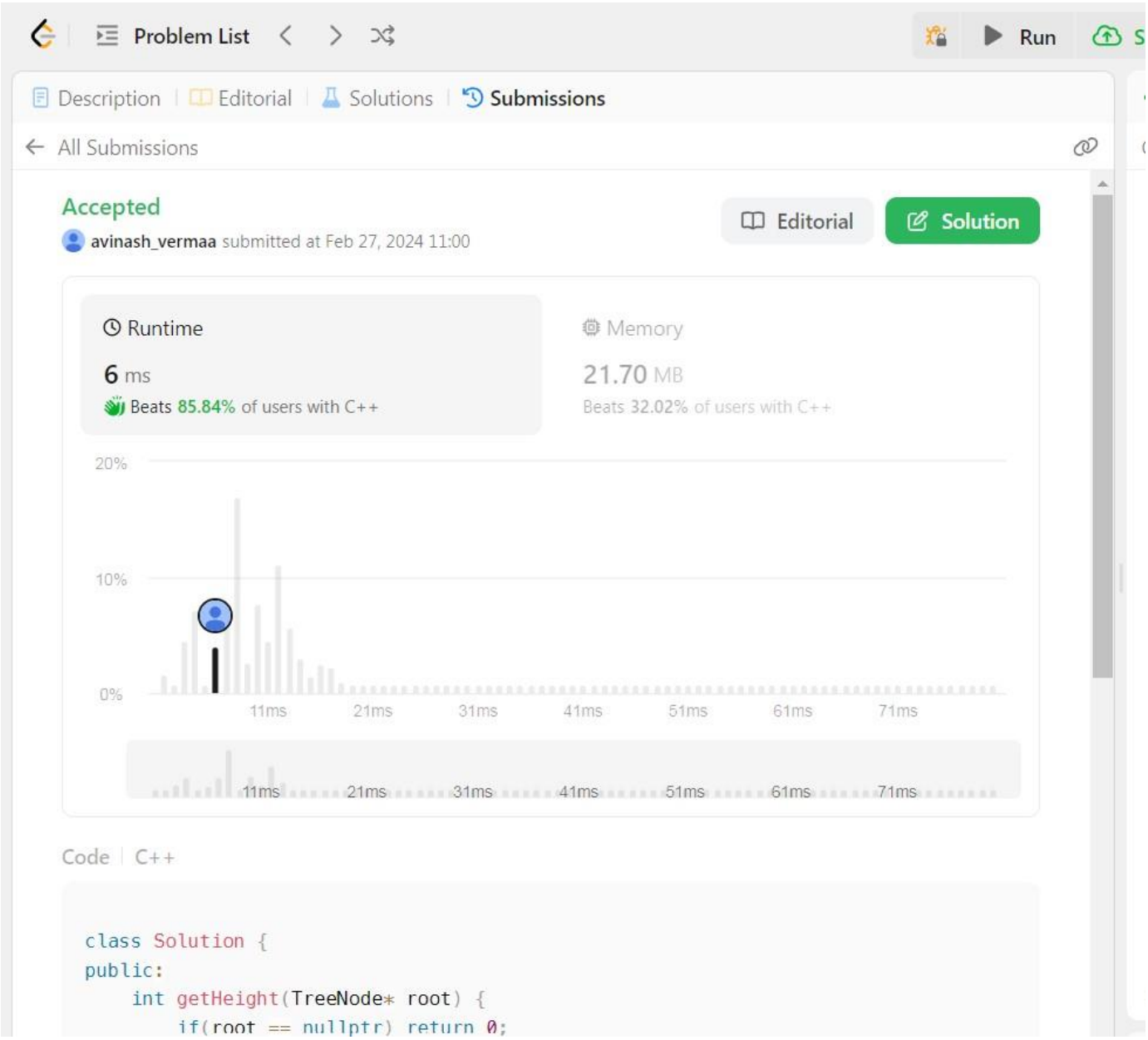
Given a binary tree, determine if it is height-balanced.

Code :

```
class Solution {
public:
   int getHeight(TreeNode* root) {
      if(root == nullptr) return 0;
      int lH = getHeight(root->left);
      int rH = getHeight(root->right);
      if(lH == -1 || rH == -1 || abs(lH - rH) > 1) return -1;
      int height = 1 + max(lH, rH);
      return height;
   }

   bool isBalanced(TreeNode* root) {
      if(root == nullptr) return true;
      if(getHeight(root) == -1) return false;
      return true;
   }
};
```

Output :


Problem List
Run
Description
Editorial
Solutions
Submissions
All Submissions
Accepted
avinash_vermaa submitted at Feb 27, 2024 11:00
Editorial
Solution
Runtime
6 ms
Beats 85.84% of users with C++
Memory
21.70 MB
Beats 32.02% of users with C++
20%
10%
0%
11ms
21ms
31ms
41ms
51ms
61ms
71ms
Code
C++
class Solution {
public:
int getHeight(TreeNode* root) {
if(root == nullptr) return 0;

Objective:

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

Code :

```
class Solution {
public:
    bool c(TreeNode* left, TreeNode* right){
        if(!left || !right){
            return left==right;
        }
        if(left->val!=right->val)return false;
        return c(left->left,right->right) && c(left->right,right->left);
    }
    bool isSymmetric(TreeNode* root) {
        return c(root,root);
    }
};
```

Output :