



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-2.1

Student Name: Aayush Gurung

Branch: CSE

Semester: 6

Subject Name: CC LAB

UID:20BCS5323

Section/Group:DM_607(A)

Date of Performance:10-03-2023

Subject Code: 20CSP-351

Aim:

To demonstrate the concept of Trees

Objective

Problem 1:

Balanced binary tree

Code:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
 * };
 */
class Solution {
public:
    int chk(TreeNode* root){
        if(root==NULL){return 0;}
        int l=chk(root->left);
        int r=chk(root->right);
        if(l==-1 || r==-1){return -1;}
        if(abs(l-r)>1){return -1;}
        return 1+max(l, r);
    }
    bool isBalanced(TreeNode* root) {
        return chk(root)!=-1;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
}  
};  
};
```

Output:

The screenshot displays the LeetCode web interface. The top navigation bar includes the LeetCode logo, a 'Problem List' button, a 'Premium' badge, and user profile icons. The main content area is divided into tabs: 'Description', 'Editorial', 'Solutions (4.6K)', and 'Submissions'. The 'Submissions' tab is active, showing a green 'Accepted' status and a 'Next question' button. Below this, there are links to '111. Minimum Depth of Binary Tree' and '104. Maximum Depth of Binary Tree'. A dropdown menu for 'All statuses' and another for 'All languages' are visible. The 'Accepted' status is confirmed with a 'C++' language selection button. The right side of the interface shows the C++ code for a solution, which defines a 'TreeNode' struct and a 'Solution' class with a 'chk' method. The code is as follows:

```
/**  
 * Definition for a binary tree node.  
 * struct TreeNode {  
 *     int val;  
 *     TreeNode *left;  
 *     TreeNode *right;  
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}  
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}  
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), l  
 * };  
 */  
class Solution {  
public:  
    int chk(TreeNode* root){  
        if(root==NULL){return 0;}  
        int l=chk(root->left);  
        int r=chk(root->right);  
        if(l== -1 || r== -1){return -1;}  
        if(abs(l-r)>1){return -1;}  
        return 1+max(l, r);  
    }  
};
```

At the bottom right, there are buttons for 'Console', 'Run', and 'Submit'.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Problem 2:

Path sum

Code:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
 * };
 */
class Solution {
public:
    bool solve(TreeNode* root,int targetSum ,int &curr){
        if(root==NULL) return false;
        curr+=root->val;
        if(curr==targetSum && root->left==NULL && root->right==NULL) return true;

        bool left =solve(root->left,targetSum,curr);
        bool right=solve(root->right,targetSum,curr);
        curr-=root->val;

        return left || right;
    }
    bool hasPathSum(TreeNode* root, int targetSum) {
        if(root==NULL) return false;
        int cr=0;
        bool ans=solve(root,targetSum,cr);
        return ans;
    }
};
```

Output:



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

DescriptionEditorialSolutions (5.6K)SubmissionsClose

Accepted

Next question

113. Path Sum II

More challenges

113. Path Sum II124. Binary Tree Maximum Path Sum

129. Sum Root to Leaf Numbers

All statusesAll languages

Accepted
a few seconds agoC++

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), l
 * };
 */
class Solution {
public:
    bool solve(TreeNode* root, int targetSum, int &curr){
        if(root==NULL) return false;
        curr+=root->val;
        if(curr==targetSum && root->left==NULL && root->right==NUL

        bool left =solve(root->left,targetSum,curr);
        bool right=solve(root->right,targetSum,curr);
        curr -=root->val;
    }
};
```

ConsoleRunSubmit