



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment 6

Student Name: Sahil

Branch: CSE

Semester: 6

Subject Name: AP LAB-II

UID:22bcs15101

Section/Group:641 A

Date of Performance:

Subject Code: 22CSP-351

1. Aim:

- a. To find and implement the maximum depth of Binary Tree.
- b. To develop an algorithm for Binary Tree Inorder traversal.

2. Objective:

To implement and analyze maximum depth of Binary Tree.

To develop an algorithm for Binary Tree Inorder traversal.

3. Implementation/Code:

a. class Solution {

public:

```
int maxDepth(TreeNode* root) {  
    if (!root) return 0;  
    int leftDepth = maxDepth(root->left);  
    int rightDepth = maxDepth(root->right);  
    return 1 + max(leftDepth, rightDepth);  
}
```

};

b. class Solution {

public:

```
vector<int> inorderTraversal(TreeNode* root) {  
    vector<int> ans;  
    in(root, ans);  
    return ans;  
}  
void in(TreeNode* root, vector<int>&ans)  
{  
    if (root == NULL)
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
return;  
in(root->left,ans);  
ans.push_back(root->val);  
in(root->right,ans);  
}  
};
```

4. Output:

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

root =
[3, 9, 20, null, null, 15, 7]

Output

3

Expected

3

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3 • Case 4

Input

root =
[1, null, 2, 3]

Output

[1, 3, 2]

Expected

[1, 3, 2]

5. Learning Outcome:

- Understand string manipulation techniques in C++.
- Implement efficient algorithms for detecting cyclic rotations.
- Apply mathematical approaches to solve missing number problems.
- Utilize standard library functions like accumulate and find.
- Enhance problem-solving skills through algorithm design and analysis.