



DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment 6

Student Name:Neetu

Branch: CSE

Semester: 6

Subject Name: AP LAB-II

UID:22BC17083

Section/Group:641/A

Date of Performance:11/03/25

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:

This screenshot shows a test result for 'Case 1'. At the top, it says 'Accepted' in green and 'Runtime: 0 ms'. Below this, there are two tabs: 'Case 1' (selected) and 'Case 2'. The 'Input' section shows 'root =' followed by '[3,9,20,null,null,15,7]'. The 'Output' section shows the value '3'. The 'Expected' section also shows the value '3'.

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

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

Output

3

Expected

3

This screenshot shows a test result for 'Case 1'. At the top, it says 'Accepted' in green and 'Runtime: 0 ms'. Below this, there are four tabs: 'Case 1' (selected), 'Case 2', 'Case 3', and 'Case 4'. The 'Input' section shows 'root =' followed by '[1,null,2,3]'. The 'Output' section shows the value '[1,3,2]'. The 'Expected' section also shows the value '[1,3,2]'.

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.