Problem | Submissions | Comments

Introduction to Trees

Medium | Accuracy: 91.08% | Submissions: 3K+ | Points: 4

Given an integer **i**. Print the **maximum number of nodes** on level i of a binary tree.

Example 1:

Input: 5
Output: 16

Example 2:

Input: 1
Output: 1

Your Task:

C++ (g++ 5.4) | Average Time: 15m | Start Timer

```
// } Driver Code Ends
// User function Template for C++

class Solution {
  public:
    int countNodes(int i) {
        // your code here
        if(i==1) return 1;
        return pow(2,i-1);
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Problem | Submissions | Comments

Binary Tree Representation

Easy | Accuracy: 91.09% | Submissions: 2K+ | Points: 2

You are given an array nodes. It contains 7 integers, which represents the value of nodes of the binary tree in level order traversal. You are also given a root of the tree which has a value equal to nodes[0].

Your task to construct a binary tree by creating nodes for the remaining 6 nodes.

Example:

Input:
nodes = [1 2 3 4 5 6 7]
Output:

C++ (g++ 5.4) | Average Time: 15m | Start Timer

```
// } Driver Code Ends
//User function Template for C++


class Solution{
public:

    void create_tree(node* root0, vector<int> &arr){
        //Your code goes here
        queue<node*> q;
        q.push(root0);

        int i = 1;
        int n = arr.size();

        while(i < n){
            node* root = q.front();
            q.pop();

            node* left = newNode(arr[i]);
            root->left = left;
            i++;
            q.push(left);

            node* right = newNode(arr[i]);
            root->right = right;
            i++;
            q.push(right);

        }
    }

};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Preorder Traversal

Basic Accuracy: 62.73% Submissions: 121K+ Points: 1

Given a binary tree, find its preorder traversal.

Example 1:

Input:

Output: 1 4 4 2

Example 2:

Input:


```
// } Driver Code Ends


//Function to return a list containing the preorder traversal of the tree.

vector <int> preorder(Node* root)
{
  // Your code here
    vector <int> res;
    if(root != NULL){
        cout << (root->data) << " ";
        preorder(root->left);
        preorder(root->right);
        return res;
    }
}
```

Problem | Editorial | Submissions | Comments

Inorder Traversal

Basic Accuracy: 67.15% Submissions: 123K+ Points: 1

Given a Binary Tree, find the In-Order Traversal of it.

Example 1:

Input:

Output: 3 1 2

Example 2:

Input:

C++ (g++ 5.4) Average Time: 15m Start Timer

```
// } Driver Code Ends


class Solution {
  public:
    // Function to return a list containing the inorder traversal of the tree.


    vector<int> inOrder(Node* root) {
        // Your code here
        vector <int> res;
        if(root != NULL){

            inOrder(root->left);
            cout << (root->data) << " ";
            inOrder(root->right);
            return res;

        }
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Problem | Editorial | Submissions | Comments

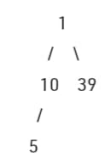
Postorder Traversal

Basic | Accuracy: 74.96% | Submissions: 89K+ | Points: 1

Given a binary tree, find the Postorder Traversal of it.

For Example, the postorder traversal of the following tree is:

5 10 39 1



Example 1:

Input:

C++ (g++ 5.4) | Average Time: 15m | Start Timer

```
// } Driver Code Ends



//Function to return a list containing the postorder traversal of the tree.
vector <int> postOrder(Node* root)
{
  // Your code here
        vector <int> res;
        if(root != NULL){

            postOrder(root->left);
            postOrder(root->right);
            cout << (root->data) << " ";

            return res;

        }
}
```

Custom Input | Compile & Run | Submit