Sum of Tree

Problem Submissions Leaderboard Discussions

Submitted 6 hours ago • Score: 20.00 Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Test Case #4	✓	Test Case #5
✓	Test Case #6	✓	Test Case #7	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11

Submitted Code

Language: C++20 Open in editor

```cpp
#include <bits/stdc++.h>
using namespace std;

class Node
{
public:
    int val;
    Node *left;
    Node *right;
    Node(int val)
    {
        this->val = val;
        this->left = NULL;
        this->right = NULL;
    }
};

Node *inputTree()
{
    int val;
    cin >> val;
    Node *root;
    if (val == -1)
        root = NULL;
    else
        root = new Node(val);

    queue<Node *> q;
    if (root)
        q.push(root);

    while (!q.empty())
    {
        Node *f = q.front();
        q.pop();

        int l, r;
        cin >> l >> r;
        Node *myLeft, *myRight;

        if (l == -1)
            myLeft = NULL;
        else
            myLeft = new Node(l);

        if (r == -1)
            myRight = NULL;
        else
            myRight = new Node(r);

        f->left = myLeft;
        f->right = myRight;

        if (f->left)
            q.push(f->left);
        if (f->right)
            q.push(f->right);
    }
    return root;
}

long long int levelOrderSum(Node *root)
{
    if(root == NULL)
    {
        return 0;
    }
    queue<Node *> q;
    q.push(root);


    long long int sum =0;
    while(!q.empty())
    {
        Node *f = q.front();
        q.pop();

        // cout<<f->val<<" ";
        sum+= f->val;

        if(f->left) q.push(f->left);
        if(f->right) q.push(f->right);
    }
    return sum;
}

int main()
{
    Node *root = inputTree();
    long long int s = levelOrderSum(root);
    cout<<s<<endl;

    return 0;
}
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |