Max Min Leaf

Submitted 5 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

```
#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;
    }
};
// global queue
queue<int> q;

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);
```

```
33
34      while (!q.empty())
35      {
36          Node *f = q.front();
37          q.pop();
38
39          int l, r;
40          cin >> l >> r;
41          Node *myLeft, *myRight;
42
43          if (l == -1)
44              myLeft = NULL;
45          else
46              myLeft = new Node(l);
47
48          if (r == -1)
49              myRight = NULL;
50          else
51              myRight = new Node(r);
52
53          f->left = myLeft;
54          f->right = myRight;
55
56          if (f->left)
57              q.push(f->left);
58          if (f->right)
59              q.push(f->right);
60      }
61      return root;
62  }
63
64  void maxMinLeaf(Node *root)
65  {
66      if (root == NULL)
67          return ;
68      if(root ->left == NULL and
69      root -> right == NULL)
70          q.push(root->val);
71      maxMinLeaf(root->left);
72      maxMinLeaf(root->right);
73
74
75
76  }
77  int main()
78  {
79      Node *root = inputTree();
80
81      maxMinLeaf(root);
82
83      int mx = INT_MIN;
84      int mn = INT_MAX;
85
86      while (!q.empty())
87      {
88          // cout<<q.front()<<" ";
89          if (q.front() >= mx)
90          {
91              mx = q.front();
92          }
93
94          if (q.front() <= mn)
95          {
96              mn = q.front();
97          }
98
```

```
        q.pop();
    }

    // cout<<"max = "<<mx<<" "<<"min = "<<mn<<endl;
    cout << mx << " " << mn << endl;

    return 0;
}
```