

PREPARE^{NEW}

CERTIFY

COMPETE

Search



yasinarafat2413 ▾

[All Contests](#) > [Assignment 04](#) | [Basic Data Structures](#) | [Batch 03](#) > [Max Min Leaf](#)

Max Min Leaf

Problem

Submissions

Leaderboard

Discussions

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

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 class Node
5 {
6 public:
7     int val;
8     Node *left;
9     Node *right;
10    Node(int val)
11    {
12        this->val = val;
13        this->left = NULL;
14        this->right = NULL;
15    }
16 };
17 // global queue
18 queue<int> q;
19
20 Node *inputTree()
21 {
22     int val;
23     cin >> val;
24     Node *root;
25     if (val == -1)
26         root = NULL;
27     else
28         root = new Node(val);
29
30     queue<Node *> q;
31     if (root)
32         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     }
```

```
99     q.pop();
100 }
101
102 // cout<<"max = "<<mx<<" "<<"min = "<<mn<<endl;
103 cout << mx << " " << mn << endl;
104
105 return 0;
106 }
```

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |