APL-INLAB-2

locked

by pmani_naga_123

You need to implement a Binary Search Tree containing integer values. You are given a total of N queries of the following types:

1) **insert** key

This query inserts key into the Binary Search Tree at the correct position. The output of this query is the level of the inserted node. Assume that the level of root node is 0.

2) **height**

The output of this query is the height of the tree. The height of the tree is the length of the longest path (in terms of number of edges) from root to the leaf.

3) **ceil** num

The output of this query is the smallest node in the tree whose value is greater than or equal to num. If no node in the tree has value greater than equal to num, output 10000.

Input Format

The first line contains an integer N denoting the total number of queries. Subsequently N lines contain one of the above mentioned queries.

Note: The first query will always be an insert query.

Constraints

$1 \leq N \leq 1000$

Insert keys are unique.

$1 \leq Key < 10000$

Output Format

Print output of each query, one per line.

Sample Input

```
7
insert 3
height
insert 5
insert 7
ceil 6
insert 2
height
```

Sample Output

```
0
0
1
2
7
1
2
```

Explanation

After inserting 3 tree contains only one root element. As level of root is 0 output is 0.

Height of the tree here is 0. Here root is also leaf. Height of leaf is 0.

After inserting 5 tree contains two elements. As the level of inserted node is 1, the output of this query is 1.

After inserting 7, the tree contains three elements. As level of inserted node is 2, the output of this query is 2.

Height of the tree now is 2.

Ceil of 6 is 7.

Inserting 2 leaves the height of the tree unchanged. The node 2 is inserted at level 1, so the output of the query is 1.

Height of the tree is 2.

Submissions: 75
Max Score: 10
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```
#include <string>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


/* Create your class(es) here
   You may use more than one class */



// ! DO NOT CHANGE WHEN SUBMITTING !
int main() {
    int N,val;
    string cmd;

    BST tree;   // BST is the class you have to create.
      // Contains atleast 3 public methods.
      // insert(val), ceil(val), height()

    cin >> N;
    for(int i=0;i<N;++i) {
   cin >> cmd;
   if(cmd == "insert") {
      cin >> val;
      cout << tree.insert(val) << "\n";
   } else if(cmd == "ceil") {
      cin >> val;
      cout << tree.ceil(val) << "\n";
   } else if(cmd == "height") {
      cout << tree.height() << "\n";
   }

    }
    return 0;
}
// ! DO NOT CHANGE UNTIL THIS LINE


```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code