Day 22: Binary Search Trees

by vatsalchanana

Problem Submissions Leaderboard Discussions

Welcome to Day 22! Check out a video review of binary search trees and heaps here, or just jump right into the problem.

The height of a binary tree is the number of nodes on the largest path from root to any leaf. You are given a pointer *root* pointing to the root of a binary search tree. Return the height of the tree.
You only have to complete the function *getHeight* given in the editor.

Input Format

First line contains *T*, the number of test cases. Next *T* lines contain an integer *data* to be added to the binary search tree.

Output Format

Output the height of the binary search tree.

Sample Input

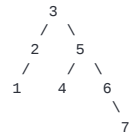
```
7
3
5
2
1
4
6
7
```

Sample Output

```
4
```

Explanation

The Binary Search tree formed with the given values is



The maximum length root to leaf path is 3->5->6->7. There are 4 nodes in this path. Therefore the height of the binary tree = 4.

Submissions: 3528
Max Score: 100
Difficulty: Easy

More

Current Buffer (saved locally, editable) C++

```
#include <iostream>
#include <cstddef>
using namespace std;
class Node{
    public:
        int data;
        Node *left,*right;
        Node(int d){
            data=d;
            left=right=NULL;
        }
};
class Solution{
    public:
        Node* insert(Node* root, int data){
            if(root==NULL){
                return new Node(data);
            }
            else{
                Node* cur;
                if(data<=root->data){
                    cur=insert(root->left,data);
                    root->left=cur;
                }
                else{
                   cur=insert(root->right,data);
                   root->right=cur;
                 }
           return root;
           }
        }
        int getHeight(Node* root){
          //Write your code here
        }

};//End of Solution
int main(){
    Solution myTree;
    Node* root=NULL;
    int T,data;
    cin>>T;
    while(T-->0){
        cin>>data;
        root= myTree.insert(root,data);
    }
    int height= myTree.getHeight(root);
    cout<<height;
    return 0;
}
```

Line: 7 Col: 1

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