Problem | Editorial | Submissions | Doubt Support | C++ (g++ 5.4) | Test against custom input

```
        20      30
       /  \
     10    15
```

Output: 0

Explanation:
Leaves 10, 15 and 30 are not at same level.

Your Task:
You dont need to read input or print anything. Complete the function **check()** which takes root node as input parameter and returns true/false depending on whether all the leaf nodes are at the same level or not.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(height of tree)

Constraints:
1 ≤ N ≤ 10^3

View Bookmarked Problems (https://practice.geeksforgeeks.org/explore/?problemType=bookmark)

Company Tags

Topic Tags

Related Courses

We are replacing the old Disqus forum with the new Discussions section given below. Click here (https://practice.geeksforgeeks.org/comments/leaf-at-same-level/1/?rel=https://practice.geeksforgeeks.org/problems/leaf-at-same-level/1) to view old Disqus comments.

Discussions (61 Threads)

Output Window

Problem Solved Successfully

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

Test Cases Passed: **1033** / 1033

Total Time Taken: **0.26** / 1.56

Correct Submission Count: **2**

Attempts No.: **4**

```
    Node* right;
};
*/

class Solution{
  public:
    void solve(Node* root, int* level, int currentLevel, bool* ans){
        if(root == NULL)
            return;

        if(root->left == NULL and root->right == NULL){
            if(*level == -1)
                *level = currentLevel;
            else if(*level != currentLevel)
                *ans = *ans and false;
            else *ans = *ans and true;
        }
        solve(root->left, level, currentLevel + 1, ans);


        solve(root->right, level, currentLevel + 1, ans);
    }
    /*You are required to complete this method*/
```

Compile & Run | Submit

sikandarburnwal408 2 days ago

Can anyone explain what is the problem in this co
***pass 8 conditions out of 1033 test cases*:**

```
bool check(Node *root)
 {
   if(root==nullptr) return true;
   map<int,Node*> mp;
   queue<pair<Node*,int>> q;
   q.push({root,0});
   while(q.empty()==false)
   {
     auto u=q.front();
     q.pop();
     Node*curr=u.first;
     int hd=u.second;
     mp[hd]=curr;
     if(curr->left!=nullptr) q.push({curr->left,hd+
     if(curr->right!=nullptr) q.push({curr->right,h
   }
   int level=-1;
   for(auto u: mp)
   {
     if(u.second->left==nullptr && u.second->rig
     {
       if(level==-1)
       {
         level=u.first;
       }
       if(u.first!=level) return false;
     }
   }
   return true;
 }
```

Reply | Open Externally

Show 1 Replies

xzqtor 5 days ago

why I'm getting TLE, I think it is O(N). Can anyone h