

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:

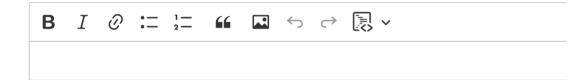
 $1 \le N \le 10^3$

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



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 08 2 days ago

Can anyone explain what is the problem in this co-You get marks only for the first correct submission if you solve the problem without pass 8 conditions out of 1033 test cases: viewing the full solution.

```
bool check (Node *foealt) ime Taken:
Test Cases Passed:
1033 / 1033
                      if(root==nullpt0.26/1056
                      map<int,Node*> mp;
                      queue<pair<Node*,int>> q;
Correct Submission Count: q.push({root,0}); Attempts No.:
```

2

```
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,r
   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
```

```
C++ (g++ 5.4)
                  Test against custom input
95
         Noαe<sup>↑</sup> right;
96 };
97 */
 98
     class Solution{
99
100
       public:
101
         void solve(Node* root, int* level, int currentLevel, bool* ans){
102
             if(root == NULL)
103
                  return;
104
105
             if(root->left == NULL and root->right == NULL){
106
                  if(*level == -1)
107
                      *level = currentLevel;
                  else if(*level != currentLevel)
108
109
                      *ans = *ans and false;
110
                  else *ans = *ans and true;
111
112
             solve(root->left, level, currentLevel + 1, ans);
113
114
115
              solve(root->right, level, currentLevel + 1, ans);
116
         /*You are required to complete this method*/
117
118
```

xzqtor 5 days ago