101. Symmetric Tree

Easy 4671 110 Add to List Share

Given a binary tree, check whether it is a mirror of itself (ie, symmetric around its center).

For example, this binary tree `[1,2,2,3,4,4,3]` is symmetric:

```
    1
   / \
  2   2
 / \ / \
3  4 4  3
```

But the following `[1,2,2,null,3,null,3]` is not:

```
    1
   / \
  2   2
   \   \
   3    3
```

Follow up: Solve it both recursively and iteratively.

Accepted 715,841 Submissions 1,516,437

Seen this question in a real interview before? Yes No

Companies

Related Topics


```cpp
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    void doswaps(TreeNode* root){
        if(!root)
            return;
        swap(root->left,root->right);
        doswaps(root->right);
        doswaps(root->left);
    }
    bool isequal(TreeNode* left, TreeNode* right){
        if(!left && !right)
            return true;
        if((!left||!right||(left->val!=right->val)))
            return false;
        return (isequal(left->left,right->left) &&isequal(right->right,left->right));
    }
    bool isSymmetric(TreeNode* root) {
        if(!root)
            return true;
        if(!root->left && !root->right)
            return true;
        if(!root->right||!root->left)
            return false;
        doswaps(root->left);
        return isequal(root->left,root->right);

    }
};
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input `[1,2,2,3,4,4,3]`

Output `true` Diff

Expected `true`