

Description Solution Discuss (999+) Submissions

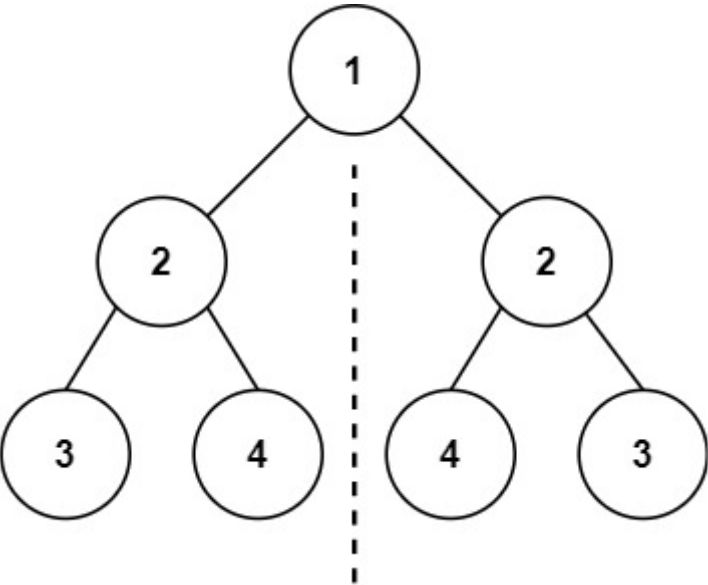
C++ Autocomplete

101. Symmetric Tree

Easy 10164 243 Add to List Share

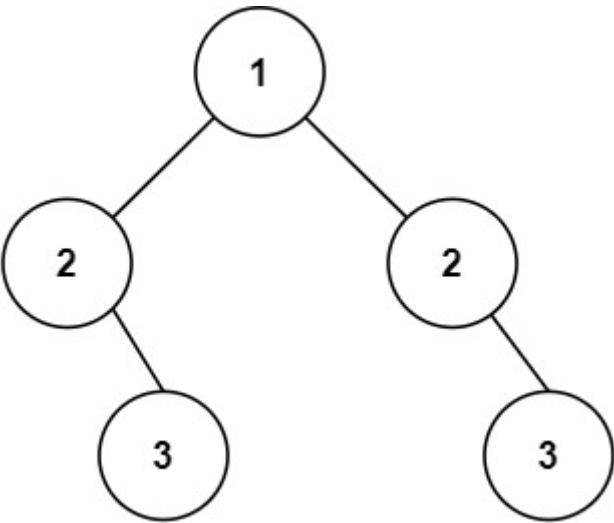
Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

Example 1:



Input: root = [1,2,2,3,4,4,3]
Output: true

Example 2:



Input: root = [1,2,2,null,3,null,3]
Output: false

Constraints:

- The number of nodes in the tree is in the range [1, 1000] .
- 100 <= Node.val <= 100

Follow up: Could you solve it both recursively and iteratively?

Accepted 1,311,776 Submissions 2,513,459

Seen this question in a real interview before? Yes No

Companies

Related Topics

```
1 /**
2  * Definition for a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     TreeNode *left;
6  *     TreeNode *right;
7  *     TreeNode() : val(0),
left(nullptr), right(nullptr) {}
8  *     TreeNode(int x) : val(x),
left(nullptr), right(nullptr) {}
9  *     TreeNode(int x, TreeNode *left,
TreeNode *right) : val(x), left(left),
right(right) {}
10  * };
11  */
12 class Solution {
13 public:
14     bool check(TreeNode* root1,
TreeNode* root2){
15         if(root1 == nullptr || root2 ==
nullptr)
16             return root1 == root2;
17         return root1->val == root2->val
and check(root1->left, root2->right)
and check(root1->right, root2->left);
18     }
19
20     bool isSymmetric(TreeNode* root) {
21         return check(root, root);
22     }
23 };
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

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

Output true Diff

Expected true