

Description

Solution

Discuss (999+)

Submissions

101. Symmetric Tree

Easy

8540

211

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:

```
graph TD; 1((1)) --- 2L((2)); 1 --- 2R((2)); 2L --- 3L((3)); 2L --- 4L((4)); 2R --- 4R((4)); 2R --- 3R((3));
```

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

Example 2:

```
graph TD; 1((1)) --- 2L((2)); 1 --- 2R((2)); 2L --- 3L((3)); 2R --- 3R((3));
```

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

i Java

Autocomplete

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10  *         this.val = val;
11  *         this.left = left;
12  *         this.right = right;
13  *     }
14  * }
15  */
16 class Solution {
17     public boolean isSymmetric(TreeNode root) {
18         if(root==null) return true;
19         return isMirror(root.left,root.right);
20     }
21     public boolean isMirror(TreeNode p, TreeNode q) {
22         if(p==null && q==null) return true;
23         if(p==null || q==null) return false;
24         return (p.val==q.val) && isMirror(p.left,q.right) &&
isMirror(p.right,q.left);
25     }
26 }
```

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

[1,2,2,3,4,4,3]

Output

true

Expected

true

Diff