

Leetcode Problem 1. (Easy)

Symmetric Tree

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].
- $-100 \leq \text{Node.val} \leq 100$
- $-10^4 \leq \text{target} \leq 10^4$

Link: <https://leetcode.com/problems/symmetric-tree/>

```
class Solution {
    public boolean isSymmetric(TreeNode root) {
        if (root == null) {
            return true;
        }
        return isMirror(root.left, root.right);
    }

    private boolean isMirror(TreeNode p, TreeNode q) {
        if (p == null && q == null) {
            return true;
        } else if (p == null || q == null) {
            return false;
        } else if (p.val != q.val) {
```

```

        return false;
    } else {
        return isMirror(p.left, q.right) && isMirror(p.right, q.left);
    }
}
}

```

LeetCode
Problem List
Premium
0

Description
Editorial
Solutions (6.8K)
Submissions
Close

Accepted
Next question
102. Binary Tree Level Order Traversal
More challenges
701. Insert into a Binary Search Tree
508. Most Frequent Subtree Sum
1202. Smallest String With Swaps
All statuses
All languages
Accepted a few seconds ago Java

Sakib Rahman
Apr 23, 2023 21:32
Details
+ Solution
Java
Runtime 0 ms
Beats 100%
Memory 40.8 MB
Beats 72.85%
Click the distribution chart to view more details
Notes
Write your notes here
Related Tags
Select tags 0/5

```

/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }

```

Console
Run
Submit