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 <= Node.val <= 100
- -10^4 <= target <= 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);
    }
}
}
```