```
# LeetCode 101: Symmetric Tree

class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right

class Solution:
    def isSymmetric(self, root: TreeNode) -> bool:
        if not root:
            return True
        return self.isMirror(root.left, root.right)

    def isMirror(self, t1: TreeNode, t2: TreeNode) -> bool:
        if not t1 and not t2:
            return True
        if not t1 or not t2:
            return False
        return (t1.val == t2.val and
                self.isMirror(t1.left, t2.right) and
                self.isMirror(t1.right, t2.left))

# Example usage:
# Tree:
#       1
#     /   \
#    2     2
#   / \   / \
#  3   4 4   3

root = TreeNode(1)
root.left = TreeNode(2, TreeNode(3), TreeNode(4))
root.right = TreeNode(2, TreeNode(4), TreeNode(3))

sol = Solution()
print("Is the tree symmetric?", sol.isSymmetric(root))
```