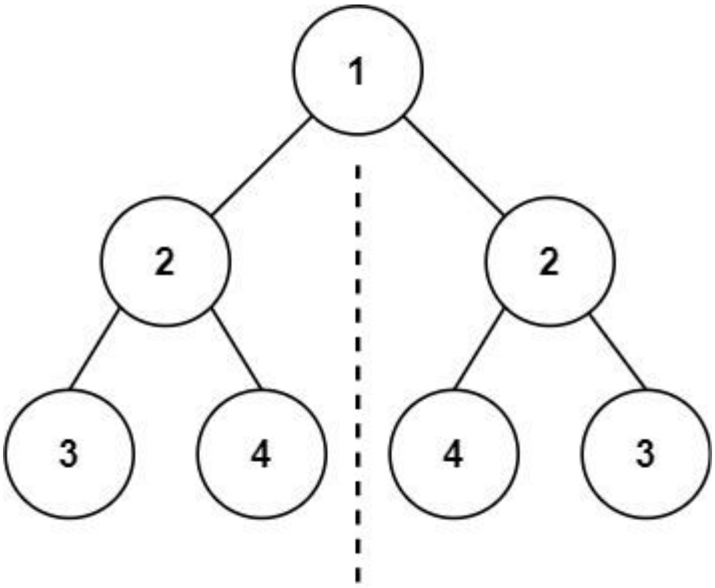
Symmetric Tree

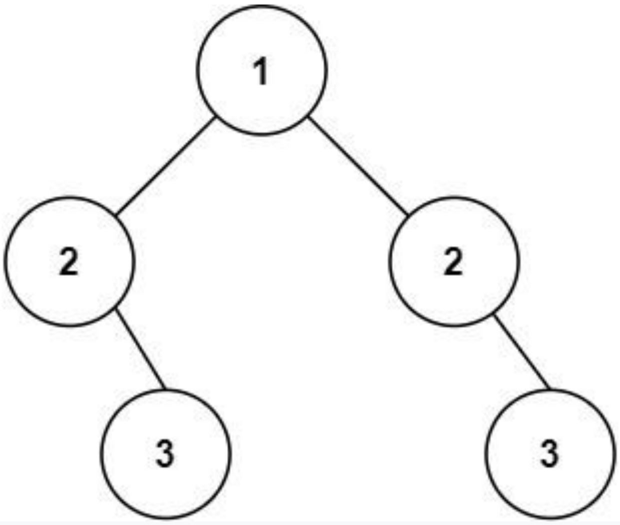
Given the `root` of a binary tree, *check whether it is a mirror of itself* (i.e., symmetric around its center).

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`

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     public int val;
 *     public TreeNode left;
 *     public TreeNode right;
 *     public TreeNode(int val=0, TreeNode left=null, TreeNode right=null) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */


public class Solution
{
    public bool IsSymmetric(TreeNode root)
    {
       return IsSame(root.left,root.right);
    }

    bool IsSame(TreeNode r1, TreeNode r2)
    {
        if((r1== null && r2!= null) || (r2== null && r1!= null) ||
            ((r1 !=null && r2 != null) && (r1.val != r2.val)))
        {
            return false;
        }

        return ((r1 == null && r2 == null) ||
                  (IsSame(r1.left, r2.right) && IsSame(r1.right,r2.left)));
    }
}
```