

```

import Foundation
// 剑指 Offer 28. 对称的二叉树
// 请实现一个函数，用来判断一棵二叉树是不是对称的。如果一棵二叉树和它的镜像一样，那么它是对称的。
//
// 例如，二叉树 [1,2,2,3,4,4,3] 是对称的。
//      1
//     / \
//    2   2
//   / \ / \
//  3  4 4  3
// 但是下面这个 [1,2,2,null,3,null,3] 则不是镜像对称的:
//      1
//     / \
//    2   2
//     \   \
//      3   3
//
// 示例 1:
// 输入: root = [1,2,2,3,4,4,3]
// 输出: true
//
// 示例 2:
// 输入: root = [1,2,2,null,3,null,3]
// 输出: false
public class TreeNode {
    public var val: Int
    public var left: TreeNode?
    public var right: TreeNode?
    public init(_ val: Int) {
        self.val = val
        self.left = nil
        self.right = nil
    }
}
class Solution {
    func isSymmetric(_ root: TreeNode?) -> Bool {
        return (root == nil) ? true: recur(root?.left, root?.right)
    }
    func recur(_ left: TreeNode?, _ right: TreeNode?) -> Bool {
        if (left == nil) && (right == nil) { return true }
        if left == nil { if right != nil { return false } }
        if right == nil { if left != nil { return false } }
        if let left = left, let right = right {
            if left.val != right.val { return false }
        }
        return recur(left?.left, right?.right) && recur(left?.right,
            right?.left)
    }
}

```