```swift
import Foundation
/// 剑指 Offer 27. 二叉树的镜像
/// 请完成一个函数，输入一个二叉树，该函数输出它的镜像。
/// 输入：root = [4,2,7,1,3,6,9]
/// 输出：[4,7,2,9,6,3,1]
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 {
    /// 执行用时：4ms, 在所有 Swift 提交中击败了 76.09% 的用户
    /// 内存消耗：13.9 MB, 在所有 Swift 提交中击败了 67.39% 的用户
    /// 通过测试用例：68 / 68
    func mirrorTree(_ root: TreeNode?) -> TreeNode? {
        if root == nil { return nil }

        let temp = root?.left
        root?.left = root?.right
        root?.right = temp

        mirrorTree(root?.left)
        mirrorTree(root?.right)
        return root
    }
}

func printTreeNode(_ root: TreeNode?) {
    var result: [Int] = []
    var cur: [TreeNode?] = [root]
    while !cur.isEmpty {
        var nex: [TreeNode?] = []
        for node in cur {
            if let node = node {
                result.append(node.val)
                if let left = node.left { nex.append(left) }
                if let right = node.right { nex.append(right) }
            }
        }
        cur = nex
    }
    print(result)
}
```