

```

import Foundation
// 剑指 Offer 55 - I. 二叉树的深度
// 输入一棵二叉树的根节点，求该树的深度。从根节点到叶节点依次经过的节点（含根、叶节点）
// 形成树的一条路径，最长路径的长度为树的深度。
//
// 例如：
// 给定二叉树 [3,9,20,null,null,15,7],
//      3
//     / \
//    9  20
//   /  \
//  15   7
// 返回它的最大深度 3 。
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 maxDepthSolution1(_ root: TreeNode?) -> Int {
        if root == nil { return 0 }
        var res = 0
        var cur: [TreeNode?] = [root]
        while !cur.isEmpty {
            res += 1
            var nex: [TreeNode?] = []
            for node in cur {
                if let node = node {
                    if let left = node.left { nex.append(left) }
                    if let right = node.right { nex.append(right) }
                }
            }
            cur = nex
        }
        return res
    }

    func maxDepth(_ root: TreeNode?) -> Int {
        if root == nil { return 0 }
        return max(maxDepth(root?.left), maxDepth(root?.right)) + 1
    }
}

```