```swift
import Foundation
/// 剑指 Offer 32 - II. 从上到下打印二叉树 II
/// 从上到下按层打印二叉树，同一层的节点按从左到右的顺序打印，每一层打印到一行。
///
/// 例如:
/// 给定二叉树: [3,9,20,null,null,15,7],
/// 返回其层次遍历结果：
/// [[3],[9,20],[15,7]]
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：8 ms, 在所有 Swift 提交中击败了 87.76% 的用户
    /// 内存消耗：13.8 MB, 在所有 Swift 提交中击败了 97.96% 的用户
    /// 通过测试用例：34 / 34
    func levelOrder(_ root: TreeNode?) -> [[Int]] {
        var results: [[Int]] = []
        var _root = root
        if _root == nil { return [] }

        var cur = [_root]
        while !cur.isEmpty {
            var nxt: [TreeNode?] = []
            var vals: [Int] = []
            for node in cur {
                if let val = node?.val {
                    vals.append(val)
                }
                if let left = node?.left {
                    nxt.append(left)
                }
                if let right = node?.right {
                    nxt.append(right)
                }
            }
            cur = nxt
            results.append(vals)
        }
        return results
   }
}
```