```swift
import Foundation
/// 剑指 Offer 32 - III. 从上到下打印二叉树 III
/// 请实现一个函数按照之字形顺序打印二叉树，即第一行按照从左到右的顺序打印，
/// 第二层按照从右到左的顺序打印，第三行再按照从左到右的顺序打印，其他行以此类推。
///
/// 例如:
/// 给定二叉树: [3,9,20,null,null,15,7],
/// 返回其层次遍历结果：
/// [[3], [20,9], [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 {
    func levelOrder(_ root: TreeNode?) -> [[Int]] {
        var results: [[Int]] = []
        var _root: TreeNode? = root
        if _root == nil { return [] }
        var cur: [TreeNode?] = [_root]
        var reverse: Bool = false
        while !cur.isEmpty {
            var nex: [TreeNode?] = []
            var vals: [Int] = []
            for node in cur.reversed() {
                if let node = node {
                    vals.append(node.val)
                    if reverse {
                        if let right = node.right { nex.append(right) }
                        if let left = node.left { nex.append(left) }
                    } else {
                        if let left = node.left { nex.append(left) }
                        if let right = node.right { nex.append(right) }
                    }
                }
            }
            reverse.toggle()
            if !vals.isEmpty { results.append(vals) }
            cur = nex
        }
        return results
   }
}
```