# [二叉树的锯齿形层次遍历](https://leetcode-cn.com/explore/interview/card/bytedance/244/linked-list-and-tree/1027/)

## 题目

给定一个二叉树，返回其节点值的锯齿形层次遍历。（即先从左往右，再从右往左进行下一层遍历，以此类推，层与层之间交替进行）。

例如：
给定二叉树 [3,9,20,null,null,15,7],

 3
 / \
 9 20
 / \
 15 7
返回锯齿形层次遍历如下：

[
 [3],
 [20,9],
 [15,7]
]

## 解题思路

1. 通过遍历每层的节点
2. 根据层数不同，用不同方式进行输出

public List<List<Integer>> zigzagLevelOrder(TreeNode root) {
 List<List<Integer>> temp = new ArrayList<>();
 all(temp, root, 0);

 List<List<Integer>> res = new ArrayList<>();
 for (int i = 0; i < temp.size(); i++) {
 if (i % 2 == 1) {
 res.add(temp.get(i));
 } else {
 Collections.reverse(temp.get(i));
 res.add(temp.get(i));
 }

 }
 return res;
}

private void all(List<List<Integer>> res, TreeNode root, int level) {
 if (root == null) {
 return;
 }

 for (int i = 0; i < level - res.size() + 1; i++) {
 res.add(new LinkedList<>());
 }

 res.get(level).add(root.val);

 all(res, root.right, level + 1);
 all(res, root.left, level + 1);
}