# 二叉树中和为某一值的路径

## 题目

[二叉树中和为某一值的路径](https://www.nowcoder.com/practice/b736e784e3e34731af99065031301bca?tpId=13&tqId=11177&rp=1&ru=%2Fta%2Fcoding-interviews&qru=%2Fta%2Fcoding-interviews%2Fquestion-ranking&tPage=2)

输入一颗二叉树的跟节点和一个整数，打印出二叉树中结点值的和为输入整数的所有路径。路径定义为从树的根结点开始往下一直到叶结点所经过的结点形成一条路径。(注意: 在返回值的 list 中，数组长度大的数组靠前)

## 解题思路

1. 将走过的路径记录下来，当走过路径总和 = target 并且当前节点是叶子节点时，该路径符合要求
2. 通过递归遍历所有可能的路径

public ArrayList<ArrayList<Integer>> FindPath(TreeNode root, int target) {
 ArrayList<ArrayList<Integer>> res = new ArrayList<>();

 FindPath(res, new LinkedList<>(), root, 0, target);

 res.sort(Comparator.comparingInt(list -> -list.size()));

 return res;
}

private void FindPath(ArrayList<ArrayList<Integer>> res,
 LinkedList<Integer> path,
 TreeNode node,
 int pathSum,
 int target) {
 if (node == null) {
 return;
 }

 if (pathSum > target) {
 return;
 }

 if (pathSum + node.val == target && node.right == null && node.left == null) {
 ArrayList<Integer> resPath = new ArrayList<>(path);
 resPath.add(node.val);
 res.add(resPath);
 return;
 }

 path.addLast(node.val);

 if (node.left != null) {
 FindPath(res, path, node.left, pathSum + node.val, target);
 }

 if (node.right != null) {
 FindPath(res, path, node.right, pathSum + node.val, target);
 }

 path.removeLast();
}