

101. Symmetric Tree



判断树是否对称(相等)

1. both null
2. either one is null
3. value of both nodes
4. recursive call the left, right subtree

104. Maximum Depth of Binary Tree



求一个二叉树的最大深度

DFS, Depth

1. base case is null
2. recursive call the left and right
3. return 1 + max(left, right)

110. Balanced Binary Tree



判断二叉树是否平衡

DFS, Depth

1. base case is null
2. same approach as **104**
3. 加一个判断语句, 如果子树已经不平衡return -1, 可以避免多余的判断 `if (Math.abs(left - right) > 1 || left == -1 || right == -1) return -1`

112. Path Sum



二叉树是否存在一条根到叶子节点的路径和等于给定值

DFS, Path sum

1. base case for null and for leaf node
2. dfs the left and right with `sum - node.value`
3. return left or right

113. Path Sum II



所有的根到叶子节点路径和等于给定值

DFS, Path sum, Backtracking

1. same approach as **112**
2. one singlepath to store current node and one allpaths as returned value
3. when hit a leaf node and `node.val == sum`, add this path to allpaths `allpaths.add(new ArrayList<>(singlepath))`
4. dfs left and right with `sum - node.val`
5. remove the last node in singlepath to **backtrack** the above level

437. Path Sum III



计算有几条父子路径和等于给定值

DFS, Path sum, Backtracking

1. same approach as **113**, except the recursive calls keep the sum unchanged
2. List singlepath to store current node value
3. use ListIterator to traversal the list from the end to beginning `ListIterator<T> itr = singlePath.listIterator(ListName.size())`
4. add the pathsum until it equals the given sum
5. count += left and right
6. remove the last element of singlepath to **backtrack** above level

543. Diameter of Binary Tree



求二叉树的最长直径(左右高度之和)

DFS, Depth

1. same approach as **104**
2. `diameter(left + right)` is different to the returned value of dfs (left or right)
3. maintain an independent value of diameter

1110. Delete Nodes And Return Forest



删除二叉树的节点并返回剩下forest的根节点

DFS, Delete

1. create a set to store to_delete values
 2. the returned value of dfs is a TreeNode
 3. dfs the root and **root.left && root.right = dfs(root.left) && dfs(root.right)**
 4. if the node appears in set and it has left or right children, add them to the forest
 5. delete the node by `node = null`
 6. return node
-