

113. Path Sum II

Description

Hints

Submissions

Discuss

Solution

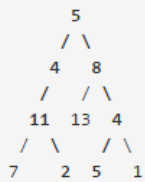
Pick One

Given a binary tree and a sum, find all root-to-leaf paths where each path's sum equals the given sum.

Note: A leaf is a node with no children.

Example:

Given the below binary tree and `sum = 22`,



Return:

```
[
  [5,4,11,2],
  [5,8,4,5]
]
```

```
public class L113 {
    public class TreeNode {
        int val;
        TreeNode left;
        TreeNode right;
        TreeNode(int x) { val = x; }
    }

    List<List<Integer>> result = new ArrayList<List<Integer>>();
    List<Integer> path = new ArrayList<>();

    public List<List<Integer>> pathSum(TreeNode root, int sum) {
        getSum(root, 0, sum);
        return result;
    }

    public void getSum(TreeNode root, int sum, int target){
        if(root == null)
            return ;
        sum += root.val;
        path.add(root.val);
        if(root.left == null && root.right == null && sum == target)
            result.add(new ArrayList<>(path)); //不能是add(path), 而必须是add(new ArrayList<>(path))
        getSum(root.left, sum, target);
        getSum(root.right, sum, target);
        sum -= root.val;
        path.remove(path.size() - 1);
        return;
    }
}
```