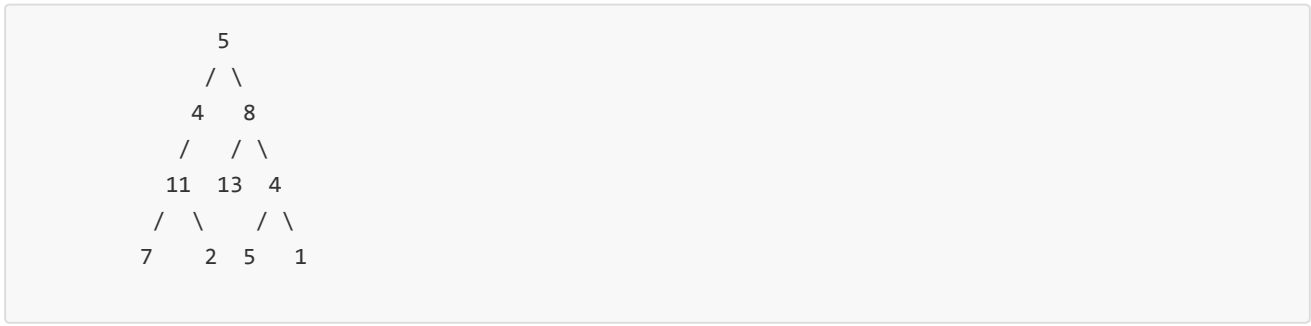
# 113. Path Sum II

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

For example,

```
sum = 22
```



return

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

记住临时vector需要及时弹出即可。

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:

    void path_dfs(TreeNode* root, int sum, vector<vector<int>>& res, vector<int> & tmp){
        if( root->left == NULL && root->right == NULL && sum == root->val ){
            tmp.push_back(root->val);
            res.push_back(tmp);
            tmp.pop_back();     // before return, pop this value
            return;
        }


        tmp.push_back(root->val );
```

```
        if( root->left != NULL ) path_dfs(root->left,sum-root->val, res, tmp);
        if( root->right != NULL ) path_dfs(root->right,sum-root->val, res, tmp);

        tmp.pop_back();         /// 记住弹出，要不然这个会越增越大的

    }

    vector<vector<int>> pathSum(TreeNode* root, int sum) {
        vector<vector<int>> res;

        if( root ==NULL ) return res;

        vector<int> tmp;
        path_dfs(root, sum, res, tmp);
        return res;


    }
};
```