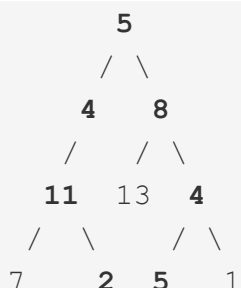


给定一个二叉树和一个目标和，找到所有从根节点到叶子节点路径总和等于给定目标和的路径。

说明: 叶子节点是指没有子节点的节点。

示例:

给定如下二叉树，以及目标和 `sum = 22`，



返回:

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

思路，该题考的不难，主要是要细心，是叶子节点的路径，那么需要把整颗树都遍历一遍，然后确认是叶子节点后，在判断是不是目标。这里我用暴力解法，遍历所有路径得到的路径。若是搜索二叉树，其和还能通过左右节点的值快速判断。

```
/**
 * 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 dfs(vector<vector<int>> &result,TreeNode* root, int sum,int num,vector<int> nums){
        num=num+root->val;
        nums.push_back(root->val);
        if(root->left==NULL&&root->right==NULL){
            if(sum==num){
                result.push_back(nums);
            }
        }else{
            if(root->left!=NULL){
                dfs(result,root->left,sum,num,nums);
            }
            if(root->right!=NULL){
                dfs(result,root->right,sum,num,nums);
            }
        }
    }
};
```

```
        }
        if (root->right != NULL) {
            dfs(result, root->right, sum, num, nums);
        }
    }
}

vector<vector<int>> pathSum(TreeNode* root, int sum) {
    vector<vector<int>> result;
    vector<int> nums;
    if (root == NULL) {
        return result;
    }
    dfs(result, root, sum, 0, nums);
    return result;
}

};
```