# 题目

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，

5

/ \

4 8

/ / \

11 13 4

/ \ / \

7 2 5 1

返回:

[

[5,4,11,2],

[5,8,4,5]

]

类似题目：剑指offer 34

# 分析

## 方法一：递归/回溯

**代码：**

vector<vector<int>> res;

vector<vector<int>> pathSum(TreeNode \*root, int targetSum)

{

vector<int> path;

dfs(root, targetSum, path);

return res;

}

void dfs(TreeNode\*root, int sum, vector<int> path)

{

if (!root)

return;

sum -= root->val;

path.push\_back(root->val);

if (!root->left && !root->right && sum == 0)

{

res.push\_back(path);

return;

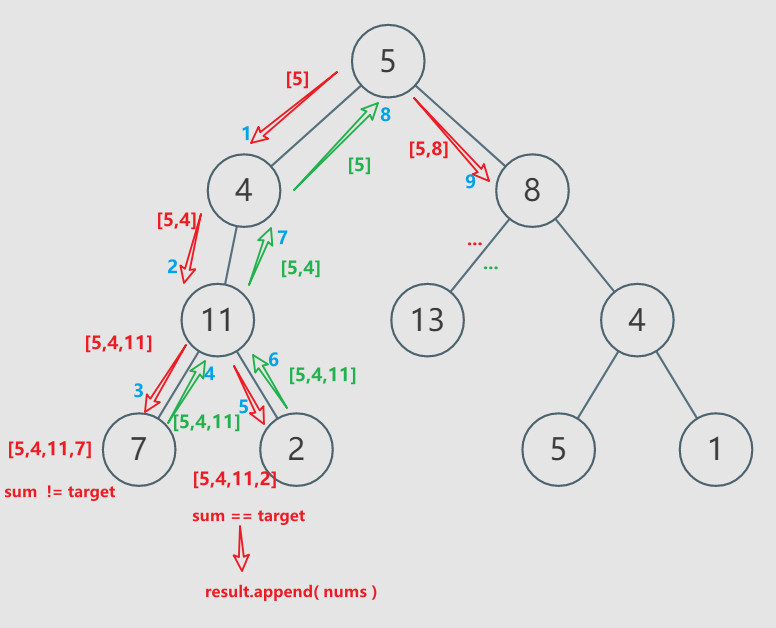
}

dfs(root->left, sum, path);

dfs(root->right, sum, path);

}

**另外一种写法：**



class Solution {

public:

vector<int> tmp;

vector<vector<int>> ret;

void dfs(TreeNode\* root, int sum)

{

int newSum = sum - root->val;

tmp.push\_back(root->val);

if(newSum==0 && !root->left && !root->right)

{

ret.push\_back(tmp);

}

if(root->left)

dfs(root->left,newSum);

if(root->right)

dfs(root->right,newSum);

tmp.pop\_back();//回溯,当前遍历的不满足则全部清除

}

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

if(root)dfs(root,sum);

return ret;

}

};

## 方法二：迭代

## 方法三：动态规划