# 题目

输入一棵二叉树和一个整数，打印出二叉树中节点值的和为输入整数的所有路径。从树的根节点开始往下一直到叶节点所经过的节点形成一条路径。

**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]

]

**提示：**

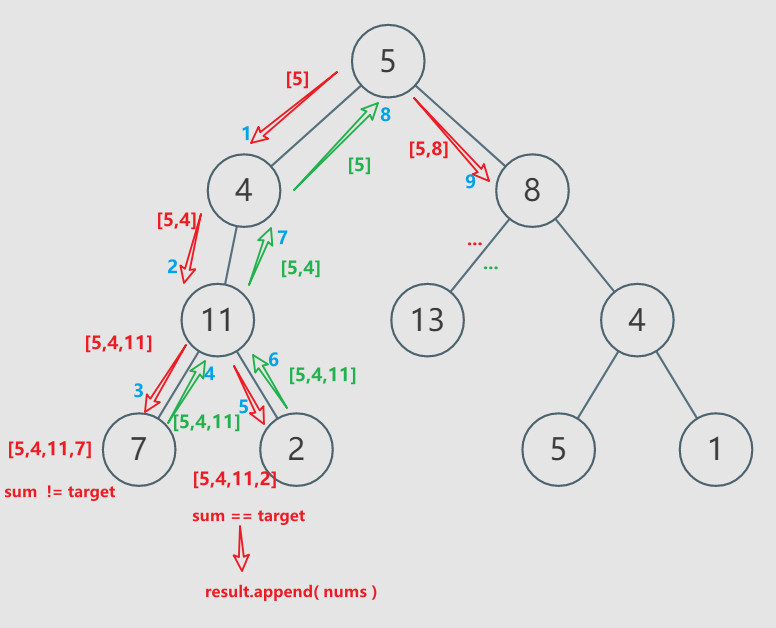
节点总数 <= 10000

**注意：**本题与主站 113 题相同：

https://leetcode-cn.com/problems/path-sum-ii/

# 分析

## 方法一：递归/回溯



/\*\*

\* 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:

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