# 题目

给定一个二叉树和一个目标和，判断该树中是否存在根节点到叶子节点的路径，这条路径上所有节点值相加等于目标和。

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 1

返回 true, 因为存在目标和为 22 的根节点到叶子节点的路径 5->4->11->2。

# 分析

## 方法一：递归法

思路：

代码：

class Solution {

public:

bool hasPathSum(TreeNode\* root, int sum) {

if (root == NULL) return false;

if (root->left == NULL && root->right == NULL &&

root->val == sum)

return true;

if (root->left != NULL &&

hasPathSum(root->left, sum - (root->val)))

return true;

if (root->right != NULL &&

hasPathSum(root->right, sum - (root->val)))

return true;

return false;

}

};

或：

class Solution {

public:

int f=0;

void dfs(TreeNode\* root,int cnt,int sum){

if(!root) return ;

if(!root->left&&!root->right){

if(sum==cnt+root->val) f=1;

return ;

}

dfs(root->left,cnt+root->val,sum);

dfs(root->right,cnt+root->val,sum);

}

bool hasPathSum(TreeNode\* root, int sum) {

if(!root) return false;

dfs(root,0,sum);

if(f) return true;

return false;

}

};

## 方法二：