

给定一个二叉树，它的每个结点都存放着一个整数值。

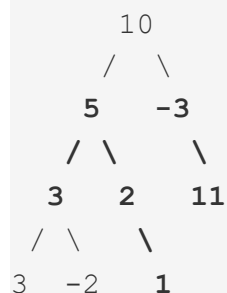
找出路径和等于给定数值的路径总数。

路径不需要从根节点开始，也不需要在叶子节点结束，但是路径方向必须是向下的（只能从父节点到子节点）。

二叉树不超过1000个节点，且节点数值范围是  $[-1000000, 1000000]$  的整数。

**示例：**

```
root = [10,5,-3,3,2,null,11,3,-2,null,1], sum = 8
```



返回 3。和等于 8 的路径有：

1. 5 -> 3
2. 5 -> 2 -> 1
3. -3 -> 11

分析，这题是典型dfs深度遍历的题，该题需要该题的难点在于切分支时，怎么不重复切出来，在这里我选择用度来标记，度为1的允许切分支，不为1的，只能向下遍历路径，路径值等于sum，则num++。

```
/**
 * 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(TreeNode* root,int sum,int val,int d,int &num){
        val=root->val+val;
        if(val==sum){
            num++;
        }
        if(root->left!=NULL){
            dfs(root->left,sum,val,d+1,num);
        }
        if(root->right!=NULL){
            dfs(root->right,sum,val,d+1,num);
        }
    }
};
```

```
        if(d==1){
            dfs(root->left,sum,0,1,num);
        }
    }
    if(root->right!=NULL){
        dfs(root->right,sum,val,d+1,num);
        if(d==1){
            dfs(root->right,sum,0,1,num);
        }
    }
}

int pathSum(TreeNode* root, int sum) {
    if(root==NULL){
        return 0;
    }
    int num=0;
    dfs(root,sum,0,1,num);
    return num;
}

};
```