## 题目描述：

## 输入一颗二叉树的跟节点和一个整数，打印出二叉树中结点值的和为输入整数的所有路径。路径定义为从树的根结点开始往下一直到叶结点所经过的结点形成一条路径。(注意: 在返回值的list中，数组长度大的数组靠前)

## 解题思路：

从跟节点开始，若跟节点的root->val=sum，且无左右子树，即为叶子节点则为一条路径。否则，递归判断左子树和右子树。以左子树为根节点，若root->left->val =sum – root->val，且无左右子树，则成功。否则继续递归。

一位数组 path 用来保存经过的各个节点，若确实为一条路径，则将该数组保存二维数组res中。

## 算法描述：

class Solution {

public:

vector<vector<int> > res;

vector<int > path;

void Find(TreeNode\* root,int sum)

{

if(root == NULL)

return;

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

if(root->left==NULL && root->right==NULL && root->val == sum)

res.push\_back(path);

else

{

if(root->left != NULL)

Find(root->left,sum - root->val);

if(root->right != NULL)

Find(root->right,sum - root->val);

}

path.pop\_back();

}

vector<vector<int> > FindPath(TreeNode\* root,int expectNumber) {

Find(root,expectNumber);

return res;

}

};