一、题目说明

题目437. Path Sum III，给定一个二叉树和整数sum，计算路径和是sum的数量，其中路径只能是从父节点向下的。难度是Easy!

二、我的解答

这个题目绝对不是Easy！最直观的想法，先判断根节点是否有路径，然后判断左子树，右子树是否有路径。

```
class Solution{
    public:
        int pathSum(TreeNode* root,int sum){
            if(root==NULL) return 0;
            return dfs(root,sum) + pathSum(root->left,sum) + pathSum(root-
>right,sum);
        }
        int dfs(TreeNode* root,int sum){
            //以root为起点，任意节点可作为结束和为sum的个数
            if(root==NULL) return 0;

            sum = sum - root->val;
            int cur = sum==0? 1: 0;

            return cur + dfs(root->left,sum) + dfs(root->right,sum);
        }
};
```

性能如下：

```
Runtime: 24 ms, faster than 49.38% of C++ online submissions for Minimum Path
Sum.
Memory Usage: 14.6 MB, less than 50.00% of C++ online submissions for Minimum
Path Sum.
```

三、优化措施

网上找了一个dp算法：这个题目类似数组求连续和，将递归算法消除一层递归!

```
class Solution{
    public:
        // dp solution
        int pathSum(TreeNode* root,int sum){
            if(root==NULL) return 0;
            sums.resize(maxDepth(root)+1,0);
            dfs(root,1,sum);
            return count;
        }
        int maxDepth(TreeNode* root){
            if(root==NULL) return 0;
            return max(maxDepth(root->left),maxDepth(root->right))+1;
        }
        void dfs(TreeNode* root,int level,int sum){
```

```
            if(root==NULL) return;

            sums[level] = sums[level-1] + root->val;
            for(int i=0;i<level;i++){
                if(sums[level]-sums[i]==sum) count++;
            }
            dfs(root->left,level+1,sum);
            dfs(root->right,level+1,sum);
        }
    private:
        int count = 0;
        vector<int> sums;
};
```

性能如下：

```
Runtime: 20 ms, faster than 77.01% of C++ online submissions for Path Sum III.
Memory Usage: 14.6 MB, less than 100.00% of C++ online submissions for Path Sum
III.
```