剑指34 二叉树中和为某一值的路径

点评：本问题是典型的二叉树方案搜索问题，使用回溯法解决，其包含 先序遍历 + 路径记录 两部分。

Tips：将target改为不断减去节点，最终用target == 0表示达到目标状态，这样能少用一个成员变量。

TODO：做一次用栈的非递归遍历试试~~~

```
// 第一次尝试，性能非常低：
// 原因大概是vector<int>作为函数参数不断传递。
class Solution {
public:
    vector<vector<int>> output;
    int mtarget;
    void DFS(TreeNode* node, int sum, vector<int> line) {
        if (!node) return;
        sum += node->val;
        line.push_back(node->val);
        if (node->left == NULL && node->right == NULL && sum == mtarget) {
            output.push_back(line);
            return;
        }
        DFS(node->left, sum, line);
        DFS(node->right, sum, line);
    }

    vector<vector<int>> pathSum(TreeNode* root, int target) {
        mtarget = target;
        DFS(root, 0, {});
        return output;
    }
};


// 自我优化：
class Solution {
public:
    vector<vector<int>> output;
    vector<int> line;
    int mtarget;
    void DFS(TreeNode* node, int sum) {
        if (!node) return;
        sum += node->val;
        line.push_back(node->val);
        if (node->left == NULL && node->right == NULL && sum == mtarget) {
            output.push_back(line);
        }
        DFS(node->left, sum);
        DFS(node->right, sum);
```

```cpp
        line.pop_back();
    }
    vector<vector<int>> pathSum(TreeNode* root, int target) {
        mtarget = target;
        DFS(root, 0);
        return output;
    }
};

// 最终改进：删除mtarget
class Solution {
public:
    vector<vector<int>> output;
    vector<int> line;
    void DFS(TreeNode* node, int sum) {
        if (!node) return;
        sum -= node->val;
        line.push_back(node->val);
        if (node->left == NULL && node->right == NULL && sum == 0) {
            output.push_back(line);
        }
        DFS(node->left, sum);
        DFS(node->right, sum);
        line.pop_back();
    }
    vector<vector<int>> pathSum(TreeNode* root, int target) {
        DFS(root, target);
        return output;
    }
};
```