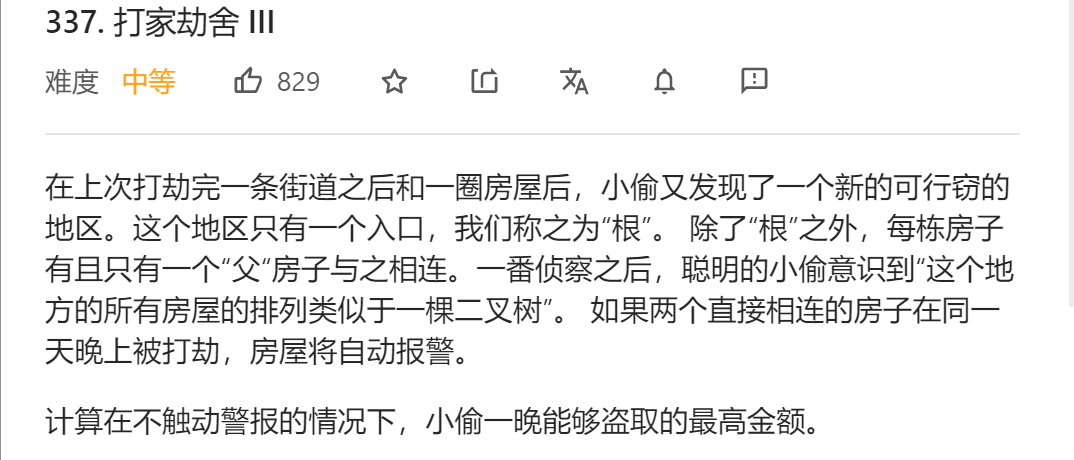
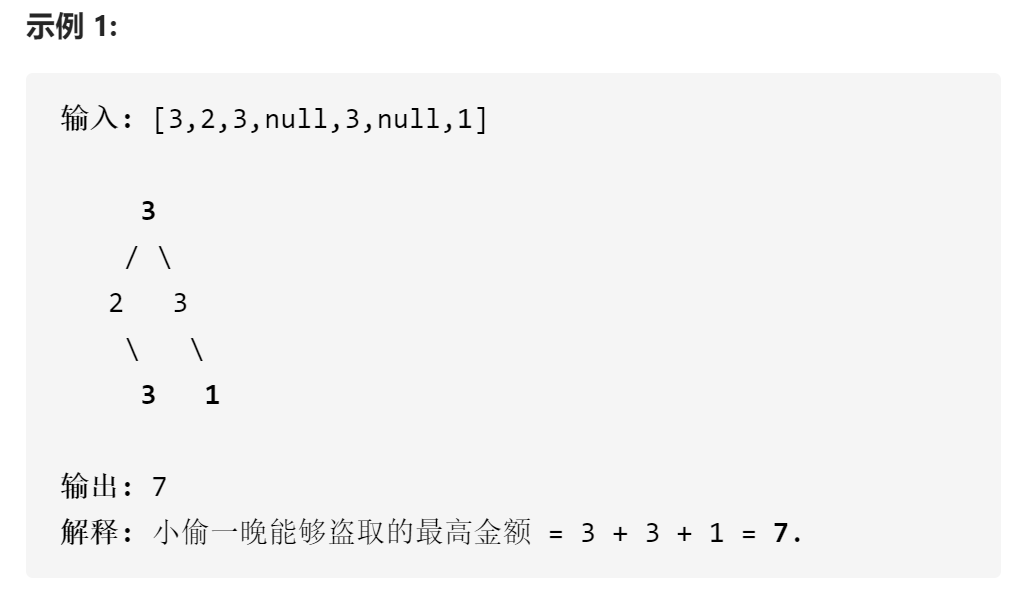
# 二叉树上的动态规划

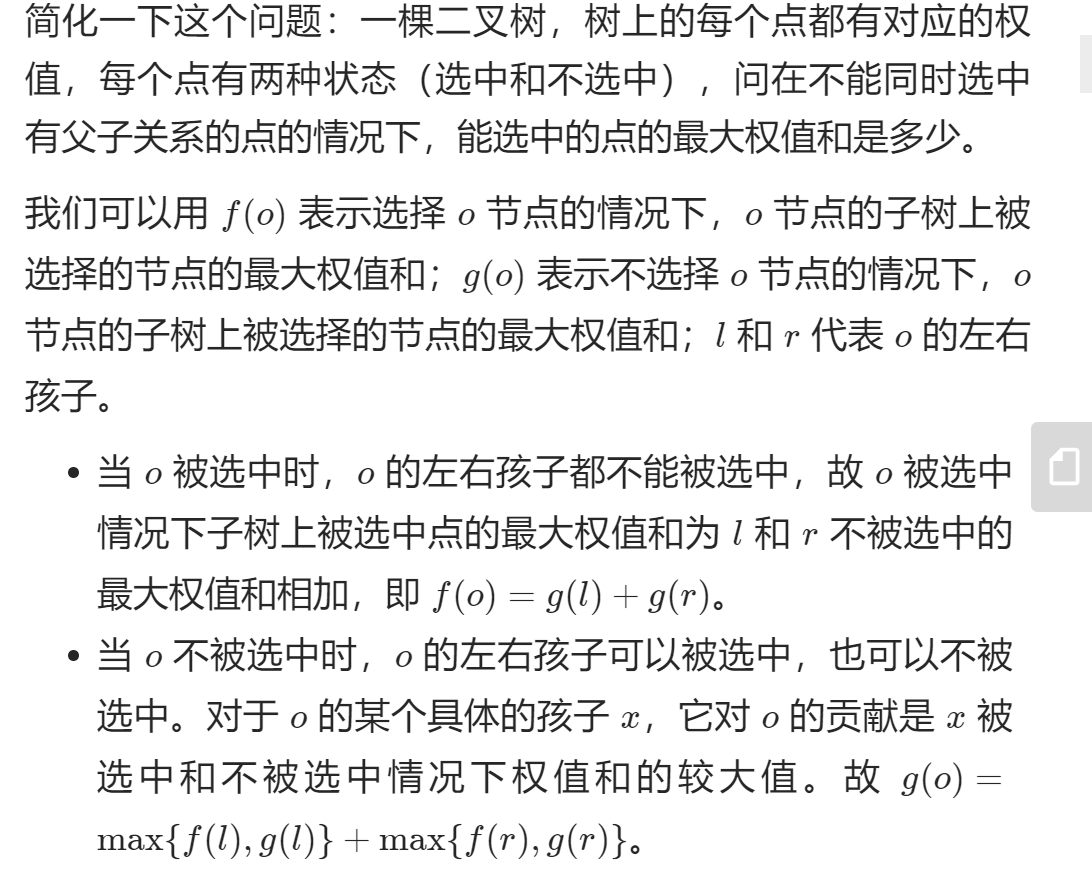
## 题目描述：





## 题目分析：

1. 我一开始认为这道题就是简单的**深度优先算法+贪心算法**，后面才发现还是我太天真了，这道题用贪心是通过不了的
2. **二叉树自带递归的属性**；
3. 动态规划的分析：



## 代码分析：

**注意后序遍历的优势性**

class Solution {  
public:  
 unordered\_map <TreeNode\*, int> f, g;  
  
 void dfs(TreeNode\* node) {  
 if (!node) {  
 return;  
 }  
 dfs(node->left);  
 dfs(node->right);  
 f[node] = node->val + g[node->left] + g[node->right];  
 g[node] = max(f[node->left], g[node->left]) + max(f[node->right], g[node->right]);  
 }  
  
 int rob(TreeNode\* root) {  
 dfs(root);  
 return max(f[root], g[root]);  
 }  
};  
  
作者：LeetCode-Solution  
链接：https://leetcode-cn.com/problems/house-robber-iii/solution/da-jia-jie-she-iii-by-leetcode-solution/  
来源：力扣（LeetCode）  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。