

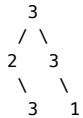
337. House Robber III

[Add to List](#)[Description \(?tab=Description\)](#)[Submission \(?tab=Submission\)](#)[Solutions \(?tab=Solutions\)](#)Total Accepted: **36287** Total Submissions: **86426** Difficulty: **Medium** Contributors: **Admin**

The thief has found himself a new place for his thievery again. There is only one entrance to this area, called the "root." Besides the root, each house has one and only one parent house. After a tour, the smart thief realized that "all houses in this place forms a binary tree". It will automatically contact the police if two directly-linked houses were broken into on the same night.

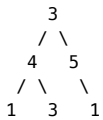
Determine the maximum amount of money the thief can rob tonight without alerting the police.

Example 1:



Maximum amount of money the thief can rob = 3 + 3 + 1 = 7.

Example 2:



Maximum amount of money the thief can rob = 4 + 5 = 9.

Credits:

Special thanks to @dietpepsi (<https://leetcode.com/discuss/user/dietpepsi>) for adding this problem and creating all test cases.

[Show Company Tags](#)[Show Tags](#)[Show Similar Problems](#)Have you met this question in a real interview? [Discuss \(https://discuss.leetcode.com/category/421\)](https://discuss.leetcode.com/category/421)[Pick One \(/problems/random-one-question/\)](/problems/random-one-question/)[Editorial Solution](#)

C++



```
1 /**
2  * Definition for a binary tree node.
3  * struct TreeNode {
4  *     int val;
5  *     TreeNode *left;
6  *     TreeNode *right;
7  *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8  * };
9  */
10 class Solution {
11 public:
12
13     // check this: https://discuss.leetcode.com/topic/39659/easy-understanding-solution-with-dfs
14     // similar idea but simpler expression
15
16     map<pair<TreeNode*, bool>, int> dp;
17     int helper(TreeNode* root, bool canInclude){
18         if(root == nullptr) return 0;
19         auto p = make_pair(root, canInclude);
20         if(dp.find(p)!=dp.end()) return dp[p];
21         if (canInclude) {
```

[✉ Send Feedback \(mailto:admin@leetcode.com?subject=Feedback\)](mailto:admin@leetcode.com?subject=Feedback)[Notes](#)

```

22         return dp[p] = max(root->val + helper(root->left, false) + helper(root->right, false),
23                             helper(root->left, true) + helper(root->right, true));
24     } else {
25         return dp[p] = helper(root->left, true) + helper(root->right, true);
26     }
27 }
28
29 int rob(TreeNode* root) {
30     if(root == nullptr) return 0;
31     int res = max(helper(root->left, true) + helper(root->right, true), root->val + helper(root->left, false) + helper
        ->right, false));
32     return res;
33 }
34

```

Custom Testcase ☐

Contribute Testcase 

Run Code

Submit Solution

[Frequently Asked Questions \(/faq/\)](#) | [Terms of Service \(/tos/\)](#)

[Privacy](#)

Copyright © 2017 LeetCode