

House Robber III

Try to solve the House Robber III problem.

We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

A thief has discovered a new neighborhood to target, where the houses can be represented as nodes in a binary tree. The money in the house is the data of the respective node. The thief can enter the neighborhood from a house represented as `root` of the binary tree. Each house has only one parent house. The thief knows that if he robs two houses that are directly connected, the police will be notified. The thief wants to know the maximum amount of money he can steal from the houses without getting caught by the police. The thief needs your help determining the maximum amount of money he can rob without alerting the police.

Constraints:

- The number of nodes in the tree is in the range $[1, 10^4]$.
- $0 \leq \text{node.data} \leq 10^4$

Examples

Sample example 1

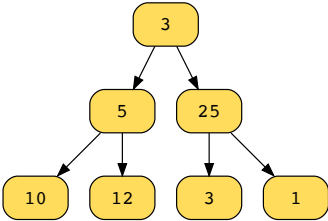
Input

3	5	25	10	12	3	1
---	---	----	----	----	---	---

The sample input is shown as the level order traversal of the binary tree. The node in yellow is the root, the nodes in blue represent the second level and the nodes in green represent the third level.

Explanation

The houses are arranged in a binary tree to represent their neighborhood relationships. The value in each node is the amount of money we can rob from that house.



```
graph TD; 3((3)) --> 5((5)); 3 --> 25((25)); 5 --> 10((10)); 5 --> 12((12)); 25 --> 3((3)); 25 --> 1((1));
```

1 of 6

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

House Robber III

1

What is the maximum amount we can rob from the following array?

`[3, 4, 5, 2, 3, null, 1]`

A) 17

B) 11

C) 10

D) 8

Submit Answer

<


Question 1 of 2
0 attempted

>

Reset Quiz ↻

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

 Drag and drop the cards to rearrange them in the correct sequence.

If the tree is empty, return 0.

Recursively calculate the maximum amount of money that can be robbed from the left and right subtrees of the `root` node.

Recursively compute the maximum amount of money that can be robbed with the

?

Tt

🔄

parent node included and
excluded both.

Return the maximum from
both the amounts computed.

Reset

Show Solution

Submit

Try it yourself

Implement your solution in `main.java` in the following coding playground.

Java



usercode > main.java

```
1 // Definition of a binary tree node class
2 // class TreeNode<T> {
3 //     T data;
4 //     TreeNode<T> left;
```

```
7 //     TreeNode(T data) {
8 //         this.data = data;
9 //         this.left = null;
10 //         this.right = null;
11 //     }
12 // }
13
14 import java.util.*;
15 import ds_v1.BinaryTree.TreeNode;
16
17 public class Main{
18     public static int rob(TreeNode<Integer> root) {
19
20         // your code will replace this placeholder return statement
21         return -1;
22     }
23 }
```

Powered by AI



Submit

Test Cases Results

Case 1

Case 2

Case 3

Input #1

[9,7,11,1,8,10,12]

House Robber III

← Back

Solution: Word Search

Next →

Solution: House Robb...

✓ Mark as
Completed



