CS 245 - Lab 7 Fall, 2021

Lab 7 - Thief in town

The goal of this assignment is to demonstrate your mastery of Tree. Download your code here.

Background

A thief found himself in a town for his new thievery journey. This town has only one entrance **root** and the whole town is in a binary tree design.

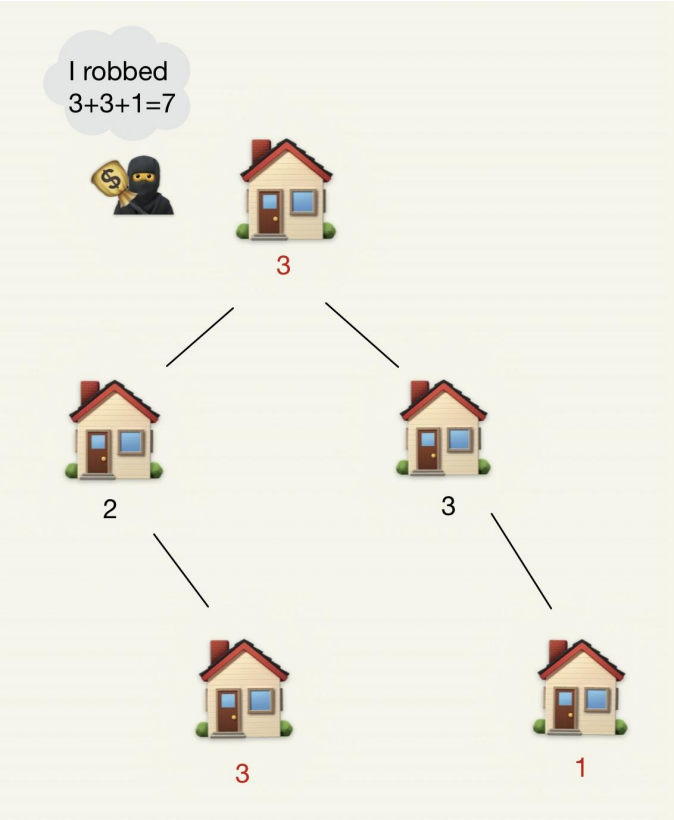
Every house has one and only one **parent** house. All of the houses have installed an alarm system that will notify the police if **two directly-linked houses were broken into on the same night**.

Your job is to find out the **maximum** number of money that the thief can rob **without** alerting the cops.

Constraints:

- *The number of nodes in the tree is in the range [1, 104].*
- *0 <= Node.val <= 104*

Examples:

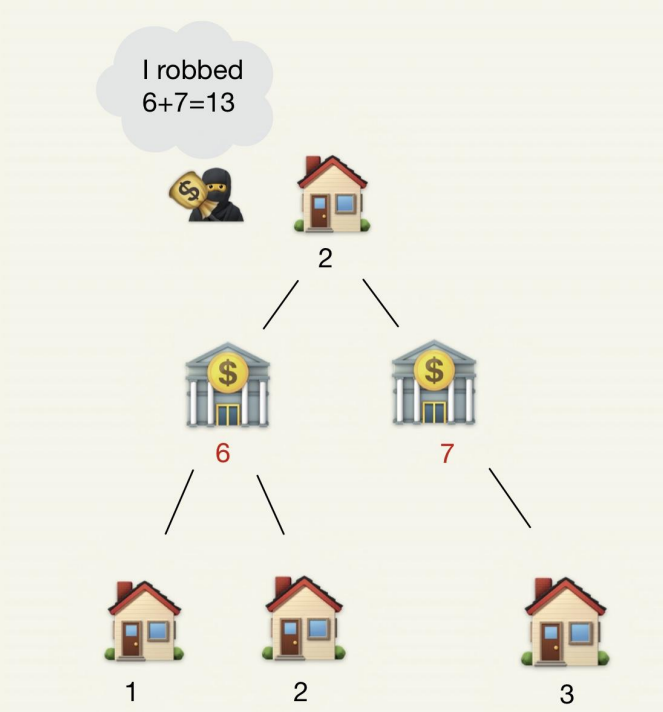



```
Input: root = [3,2,3,null,3,null,1]/[3,2,3,0,3,0,1]
```

NOTE: **null** here means there is no child node in the right or left, when you insert the node as null, enter 0.

```
Output: 7
```

```
Explanation: The maximum amount of the money the thief can
rob is 3 + 3 + 1 = 7.
```



Input: costs = [2,6,7,1,2,null,3]

Output: 13

Explanation: The maximum amount of the money that the thief can rob is 6 + 7 = 13.

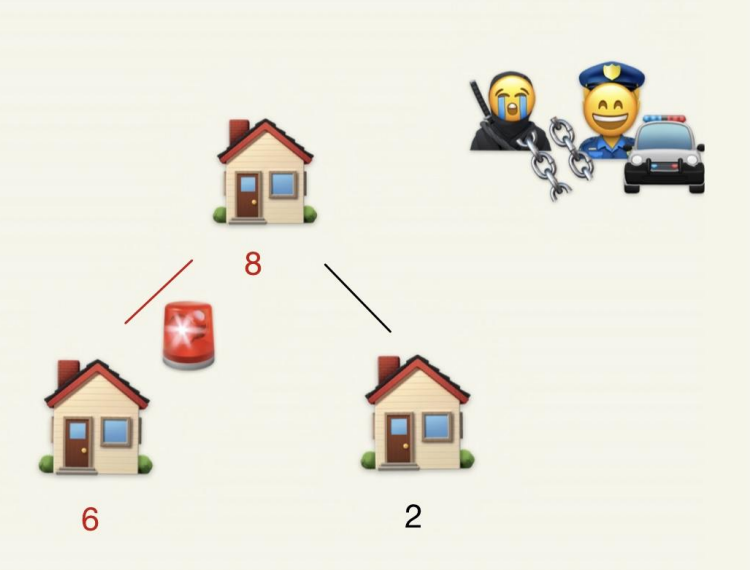


Illustration of the thief can’t rob two directly-linked houses

Requirements

- You must submit a single Java file containing the class named **HouseRobbing**. You class must contain the function named "`rob`" with the following header:

```
public int rob(TreeNode root)
```

- The "null" you see is just a placeholder to represent that the child of the parent node is missing, you don't actually set the children to null when you are passing a tree for testing. You can implement helper functions as needed.
- Hint given in the javadoc, but it is optional to use it.

Submission

- Download the source code for your implementation from Github Classroom. If you don't have a Github account, create one via github.com > Sign up.
- Select your name from the list of names available, which will link your GitHub ID. If you do not see your name on the list, speak with the instructor or one of the teaching assistants.
- Submit the repository link on Canvas. You may also add any comments in a README file (text, PDF or Word document) to help the grader understand or execute your implementation.

Grading

Your grade for this assignment will be determined as follows:

- 75% = Implementation: your class implementations must run successfully with the source files and data provided. It must produce the expected results, a sample of which appears in the Implementation section above. Any deviation from the expected results results in 0 credit for implementation.
- 15% = Decomposition: in the eyes of the grader, the degree to which your solution follows the suggestions above or otherwise represents a reasonable object-oriented and procedural decomposition to this problem.
- 10% = Style: your code must be readable to the point of being self-documenting; in other words, it must have consistent comments describing the purpose of each class and the purpose of each function within a class. Names for variables and functions must be descriptive, and any code which is not straightforward or is in any way difficult to understand must be described with comments.