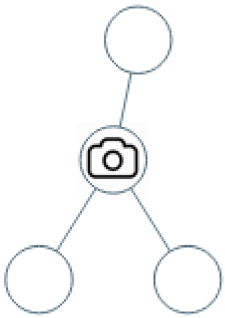
968. Binary Tree Cameras

You are given the root of a binary tree. We install cameras on the tree nodes where each camera at a node can monitor its parent, itself, and its immediate children.

Return *the minimum number of cameras needed to monitor all nodes of the tree*.

Example 1:

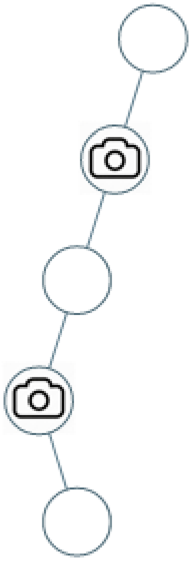


Input: root = [0,0,null,0,0]

Output: 1

Explanation: One camera is enough to monitor all nodes if placed as shown.

Example 2:



Input: root = [0,0,null,0,null,0,null,null,0]

Output: 2

Explanation: At least two cameras are needed to monitor all nodes of the tree. The above image shows one of the valid configurations of camera placement.

Constraints:

- The number of nodes in the tree is in the range [1, 1000].
- Node.val == 0

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def minCameraCover(self, root):
        self.res = 0
        def dfs(root):
            if not root:
                return 2
            l, r = dfs(root.left), dfs(root.right)
            if l == 0 or r == 0:
                self.res += 1
                return 1
            if l == 1 or r == 1:
                return 2
            else:
                return 0
        return (dfs(root) == 0)+self.res
```