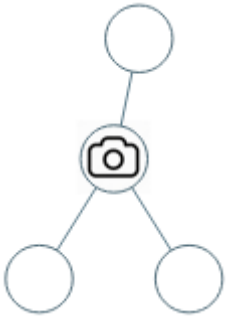
968. Binary Tree Cameras

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.

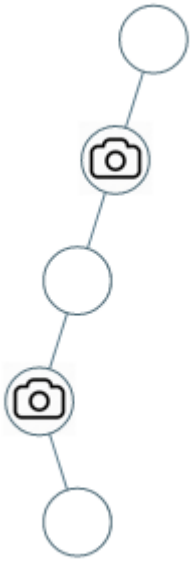
Return *the minimum number of cameras needed to monitor all nodes of the tree*.

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`

```
def minCameraCover(self, root: Optional[TreeNode]) -> int:
        if root is None:
            return 0
        cameras = [0]
        if self.numberOfCameras(root,cameras)==-1:
            cameras[0] = cameras[0]+1
        return cameras[0]

    def numberOfCameras(self,root,cameras):
        if root is None:
            return 1
        leftCam = self.numberOfCameras(root.left,cameras)
        rightCam = self.numberOfCameras(root.right,cameras)

        if leftCam == -1 or rightCam==-1:
            cameras[0] = cameras[0]+1
            return 0
        if leftCam == 0 or rightCam ==0:
            return 1
        return -1
```