BST의 각 노드를 현재값보다 더 큰 값을 가진 모든 노드의 합으로 만들어라.

Input: root = [4,1,6,0,2,5,7,null,null,null,3,null,null,null,8]
Output: [30,36,21,36,35,26,15,null,null,null,33,null,null,null,8]

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
```

1.중위 순회

```
class Solution:
    def bstToGst(self, root: TreeNode) -> TreeNode:
        def inOrder(node, val):
            if node == None:
                return val
            node.val += inOrder(node.right, val)
            val = inOrder(node.left, node.val)
            return val

        inOrder(root, 0)
        return root
```

```
class Solution:
    val = 0
    def bstToGst(self, root: TreeNode) -> TreeNode:
        if root:
            self.bstToGst(root.right)
            self.val += root.val
            root.val = self.val
            self.bstToGst(root.left)
        return root
```