

1038. Binary Search Tree to Greater Sum Tree

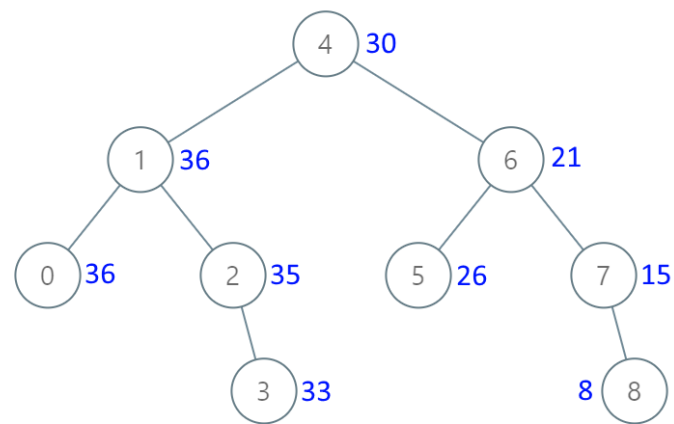
Medium 845 101 Add to List Share

Given the root of a binary **search** tree with distinct values, modify it so that every `node` has a new value equal to the sum of the values of the original tree that are greater than or equal to `node.val`.

As a reminder, a *binary search tree* is a tree that satisfies these constraints:

- The left subtree of a node contains only nodes with keys **less than** the node's key.
- The right subtree of a node contains only nodes with keys **greater than** the node's key.
- Both the left and right subtrees must also be binary search trees.

Example 1:



Input: [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]

Constraints:

```
1  /**
2   * Definition for a binary tree node.
3   * public class TreeNode {
4   *     int val;
5   *     TreeNode left;
6   *     TreeNode right;
7   *     TreeNode() {}
8   *     TreeNode(int val) { this.val = val; }
9   *     TreeNode(int val, TreeNode left, TreeNode right) {
10    *         this.val = val;
11    *         this.left = left;
12    *         this.right = right;
13    *     }
14    * }
15    */
16    class Solution {
17        public TreeNode bstToGst(TreeNode root) {
18
19        }
20    }
```