**Justin\_Tree\_DepthfirstSearch\_1038. Binary Search Tree to Greater Sum Tree**

**Concept:**

有右子樹，則加上 dfs(node.right, diff)

有左子樹，則找 dfs(node.left, node.val)

**Code:**

# 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

class Solution:

def bstToGst(self, root: TreeNode) -> TreeNode:

def dfs(node, diff):

if node.right:

rdiff = dfs(node.right, diff)

node.val += rdiff

else:

node.val += diff

if node.left:

left = dfs(node.left, node.val)

else:

return node.val

return left

if(root != None):

dfs(root, 0)

return root