**Justin\_LinkedList\_DepthfirstSearch\_0426. Convert Binary Search Tree to Sorted Doubly Linked List**

**Concept:**

利用 DFS 方法找出正確排序，並把相鄰的數相連

最後再把頭尾戶相連接

**Code:**

"""

# Definition for a Node.

class Node:

def \_\_init\_\_(self, val, left=None, right=None):

self.val = val

self.left = left

self.right = right

"""

class Solution:

def treeToDoublyList(self, root: 'Node') -> 'Node':

def helper(node):

nonlocal last, first

if node:

helper(node.left)

if last:

last.right = node

node.left = last

else:

first = node

last = node

helper(node.right)

if not root:

return None

first, last = None, None

helper(root)

last.right = first

first.left = last

return first