**Justin\_Tree\_0889.**  **Construct Binary Tree from Preorder and Postorder Traversal**

**Concept:**

利用 recursion，把 L、root、R 分出來

root 為 pre[0]

L 為 pre[1] 在 post.index 的位置+1

**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 constructFromPrePost(self, pre: List[int], post: List[int]) -> TreeNode:

if not pre:

return None

root = TreeNode(pre[0])

if len(pre) == 1:

return root

L = post.index(pre[1]) + 1

#print(L)

root.left = self.constructFromPrePost (pre[1:L+1], post[:L])

root.right = self.constructFromPrePost (pre[L+1:], post[L:-1])

return root