Construct Binary Search Tree from Preorder Traversal

class Solution:

def bstFromPreorder(self, preorder: List[int]) -> TreeNode:

def helper(lower = float('-inf'), upper = float('inf')):

nonlocal idx

# if all elements from preorder are used

# then the tree is constructed

if idx == n:

return None

val = preorder[idx]

# if the current element

# couldn't be placed here to meet BST requirements

if val < lower or val > upper:

return None

# place the current element

# and recursively construct subtrees

idx += 1

root = TreeNode(val)

root.left = helper(lower, val)

root.right = helper(val, upper)

return root

idx = 0

n = len(preorder)

return helper()