Binary tree preorder & inorder transversal

# Definition for a binary tree node.

# class TreeNode(object):

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution(object):

def buildTree(self, preorder, inorder):

"""

:type preorder: List[int]

:type inorder: List[int]

:rtype: TreeNode

"""

def check(val,inorder):

return inorder.index(val)

if not preorder or not inorder or len(preorder)!=len(inorder):

return

def rec(preorder,inorder):

ret = None

if inorder:

ret = TreeNode(preorder[0])

ind = check(preorder.pop(0),inorder)

ret.left = rec(preorder,inorder[:ind])

ret.right = rec(preorder,inorder[ind+1:])

return ret

return rec(preorder,inorder)