PROBLEM

Construct Binary Tree from Inorder and Postorder

Medium Accuracy: 64.78% Submissions: 54K+ Points: 4

Given **inorder** and **postorder** traversals of a binary tree(having **n** nodes) in the arrays **in**[] and **post**[] respectively. The task is to construct a binary tree from these traversals.

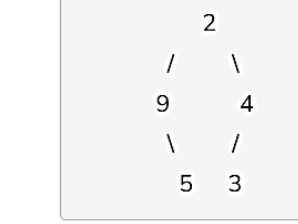
Driver code will print the preorder traversal of the constructed tree.

Example 1:

Input:
n = 8
in[] = {4, 8, 2, 5, 1, 6, 3, 7}
post[] = {8, 4, 5, 2, 6, 7, 3, 1}
Output:
1 2 4 8 5 3 6 7
Explanation:
For the given postorder and inorder traversal of tree the resultant binary tree will be

Example 2:

Input:
n = 5
in[] = {9, 5, 2, 3, 4}
post[] = {5, 9, 3, 4, 2}
Output:
2 9 5 4 3
Explanation:
The resultant binary tree will be



Your Task:

You do not need to read input or print anything. Complete the function **buildTree()** which takes the inorder, postorder traversals and the number of nodes in the tree as input parameters and returns the root node of the newly constructed Binary Tree.

Expected Time Complexity: $O(n^2)$

Expected Auxiliary Space: O(n)

Constraints:

$1 <= n <= 10^3$

$1 <= in[i], post[i] <= 10^6$

CODE

```
#User function Template for python3

'''
class Node:
        def __init__(self, data):
           self.data = data
           self.left = self.right = None
'''

#Function to return a tree created from postorder and
inoreder traversals.
class Solution:
   def buildTree(self,In, post, n):
      if n == 1:
         newNode = Node(In[0])
         return newNode

      if n == 0:
         return None

      temp = post[-1]
      pos = In.index(temp)
      root = Node(temp)
      inleft = In[:pos]
      inright = In[pos+1:]
      postleft = post[:pos]
      postright = post[pos:-1]
      root.left = self.buildTree(inleft, postleft, len(inleft))
      root.right = self.buildTree(inright, postright,
len(inright))
      return root
```