Construct tree from Inorder and LevelOrder

Given inorder and level-order traversals of a Binary Tree, construct the Binary Tree and return the root Node.

Input:
First line consists of T test cases. First line of every test case consists of N, denoting number of elements is respective arrays. Second and third line consists of arrays containing Inorder and Level-order traversal respectively.

Output:
Single line output, print the preOrder traversal of array.

Constraints:
1<=T<=100
1<=N<=100

Example:
Input:
2
3
1 0 2
0 1 2
7
3 1 4 0 5 2 6
0 1 2 3 4 5 6
Output:
0 1 2
0 1 3 4 2 5 6

```
def buildTree(level, ino):
    #code here
    #return root of tree
    return buildTreeHelper(level,ino)



def buildTreeHelper(level,inorder):
    if len(inorder)==0:
        return None
    node = Node(level[0])
    idx = inorder.index(level[0])
    hasSet = set()
    leftSet = []
    rightSet = []
    i = 0
    while i<idx:
        hasSet.add(inorder[i])
        i = i+1
    for ele in level[1:]:
        if ele in hasSet:
            leftSet.append(ele)
        else:
            rightSet.append(ele)
    node.left = buildTreeHelper(leftSet,inorder[:idx])
    node.right = buildTreeHelper(rightSet,inorder[idx+1:])
    return node
```