

Construct Binary tree

Question: Given inorder and postorder traversal of a tree, construct the binary tree.

Note: You may assume that duplicates do not exist in the tree.

Solutions:

```
class TreeNode:
```

```
    def __init__(self, x):
```

```
        self.val = x
```

```
        self.left = None
```

```
        self.right = None
```

```
class Solution:
```

```
    # @param inorder, a list of integers
```

```
    # @param postorder, a list of integers
```

```
    # @return a tree node
```

```
    def buildTree(self, inorder, postorder):
```

```
        if not inorder: return None # inorder is empty
```

```
        self.inorder, self.postorder = inorder, postorder
```

```
        return self.dfs(0, 0, len(inorder))
```

```
    def dfs(self, inLeft, postLeft, Len):
```

```
        if Len <= 0:
```

```
            return None
```

```
        root = TreeNode(self.postorder[postLeft + Len - 1])
```

```
rootPos = self.inorder.index(self.postorder[postLeft + Len - 1])
root.left = self.dfs(inLeft, postLeft, rootPos - inLeft)
root.right = self.dfs(rootPos + 1, postLeft + rootPos - inLeft, Len - 1 -
(rootPos - inLeft))
return root
```

```
Solution().buildTree([1,3,2],[3,2,1])
```