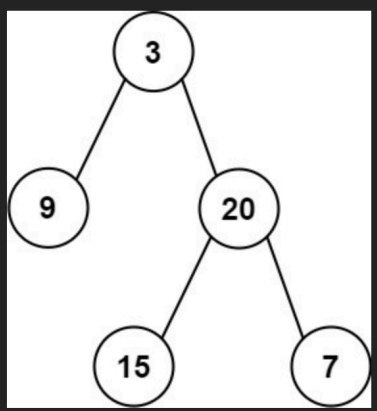
# 106. Construct Binary Tree from Inorder and Postorder Traversal

Medium | Topics | Companies

Given two integer arrays `inorder` and `postorder` where `inorder` is the inorder traversal of a binary tree and `postorder` is the postorder traversal of the same tree, construct and return *the binary tree*.

**Example 1:**




> **Input:** inorder = [9,3,15,20,7], postorder = [9,15,7,20,3]
> **Output:** [3,9,20,null,null,15,7]

```python
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def buildTree(self, inorder: List[int], postorder: List[int]) -> Optional[TreeNode]:
        # postOrder[-1] -> root

        def helper(l: int, r: int) -> TreeNode:
            if l > r:
                return None

            val = postorder.pop()
            root = TreeNode(val)

            ind = ind_map[val]

            root.right = helper(ind + 1, r)
            root.left = helper(l, ind - 1)

            return root

        ind_map = {v: i for i, v in enumerate(inorder)}

        return helper(0, len(postorder) - 1)

```

helper take left index an right index to create the root of the interval

ind_map create a map for (val, inorder_index), will always know the root val from postorder[-1], need the mapped index to separate inorder list