

PreOrder Morris Traversal

```
def preorderTraversal(self, root: Optional[TreeNode]) -> List[int]:
    if root is None:
        return []
    curr = root
    ans = []
    while curr!=None:
        leftNode = curr.left
        if leftNode is None:
            ans.append(curr.val)
            curr = curr.right
        else:
            rightMost = self.getRightMost(leftNode, curr)

            if rightMost.right is None:
                rightMost.right = curr
                ans.append(curr.val)
                curr = curr.left
            else:
                rightMost.right = None
                curr = curr.right
    return ans

def getRightMost(self, leftNode, curr):
    while leftNode.right is not None and leftNode.right is not curr:
        leftNode = leftNode.right
    return leftNode
```