94. Binary Tree Inorder Traversal

Given the `root` of a binary tree, return *the inorder traversal of its nodes' values*.

```
class Pair:
    def __init__(self,node,state):
        self.node = node
        self.state = state
class Solution:
    def inorderTraversal(self, root: TreeNode) -> List[int]:
        if root is None:
            return
        pair = Pair(root,1)
        stack = [pair]
        res = []
        while len(stack)>0:
            temp= stack[-1]
            if temp.state==1:
                temp.state = temp.state +1
                if temp.node.left:
                    stack.append(Pair(temp.node.left,1))
            elif temp.state==2:
                res.append(temp.node.val)
                temp.state = temp.state +1
                if temp.node.right:
                    stack.append(Pair(temp.node.right,1))
            else:
                stack.pop()
        return res
```

Approach2:

```
def inorderTraversal(self, root: TreeNode) -> List[int]:
        stack = []
        res = []
        curr = root
        while curr!=None or len(stack)>0:
            while curr!=None:
                stack.append(curr)
                curr = curr.left
```

```
            curr = stack.pop()
            res.append(curr.val)
            curr = curr.right
        return res
```