94. Binary Tree Inorder Traversal

link

注意循环条件`current != null || !stack.isEmpty()`

```
    public List<Integer> inorderTraversal1(TreeNode root) {
        List<Integer> res = new ArrayList();
        if(root == null)
            return res;
        helper(root,res);
        return res;
    }
    public void helper(TreeNode node, List res){
        if(node == null) return;
        helper(node.left,res);
        res.add(node.val);
        helper(node.right,res);
    }
    public List<Integer> helper(TreeNode root, res) {
        if(root == null) return res;
        Stack<TreeNode> stack = new Stack();
        TreeNode current = root;
//这里注意循环条件 current!=null 或者stack不为空，因为current.right为空的话就要看stack是不是
    空决定是否循环
        while(current != null || !stack.isEmpty()){
            while(current != null){
                stack.push(current);
                current = current.left;
            }
            current = stack.pop();
            res.add(current.val);
// if(current.right != null)写的时候这里多了一个if导致死循环
            current = current.right;
        }
        return res;
    }
```