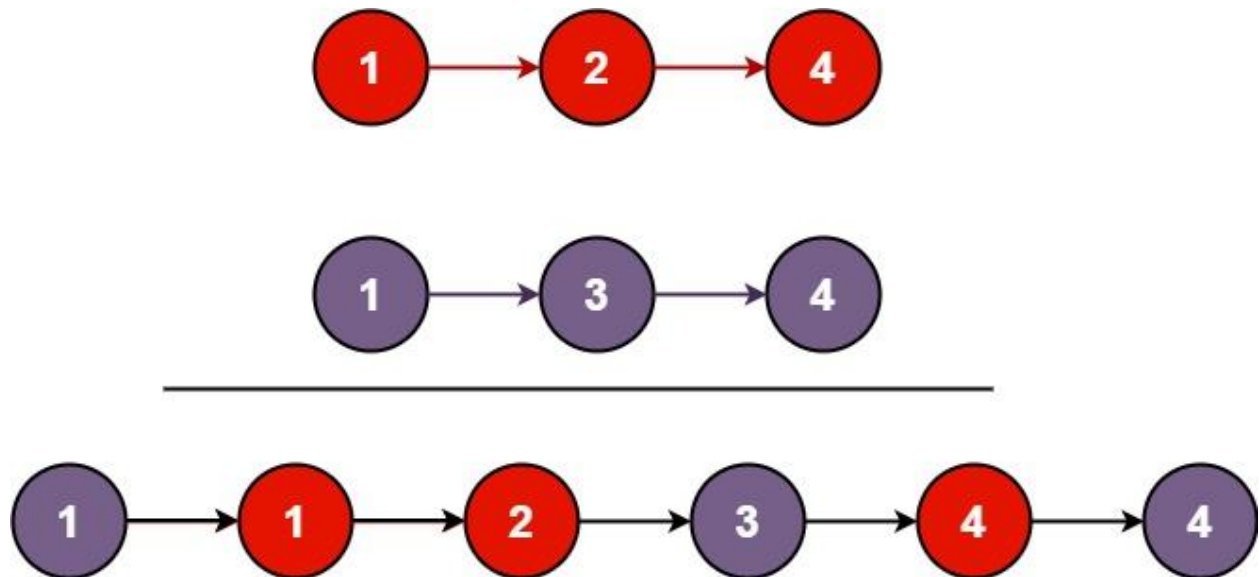


You are given the heads of two sorted linked lists list1 and list2.

Merge the two lists into one sorted list. The list should be made by splicing together the nodes of the first two lists.

Return *the head of the merged linked list*.

Example 1:



Input: list1 = [1,2,4], list2 = [1,3,4]

Output: [1,1,2,3,4,4]

Example 2:

Input: list1 = [], list2 = []

Output: []

Example 3:

Input: list1 = [], list2 = [0]

Output: [0]

Constraints:

- The number of nodes in both lists is in the range [0, 50].
- $-100 \leq \text{Node.val} \leq 100$
- Both list1 and list2 are sorted in non-decreasing order.

Solution:

```
class Solution {
    public ListNode mergeTwoLists(ListNode list1, ListNode list2) {
        ListNode head=new ListNode(0);
        ListNode p1=list1,p2=list2,p3=head;
        while(p1!=null&& p2!=null){
            if(p1.val<=p2.val){
                p3.next=p1;
                p1=p1.next;
            }else{
                p3.next=p2;
                p2=p2.next;
            }
            p3=p3.next;
        }
        if(p1!=null){
            p3.next=p1;
        }
        if(p2!=null){
            p3.next=p2;
        }
        return head.next;
    }
}
```