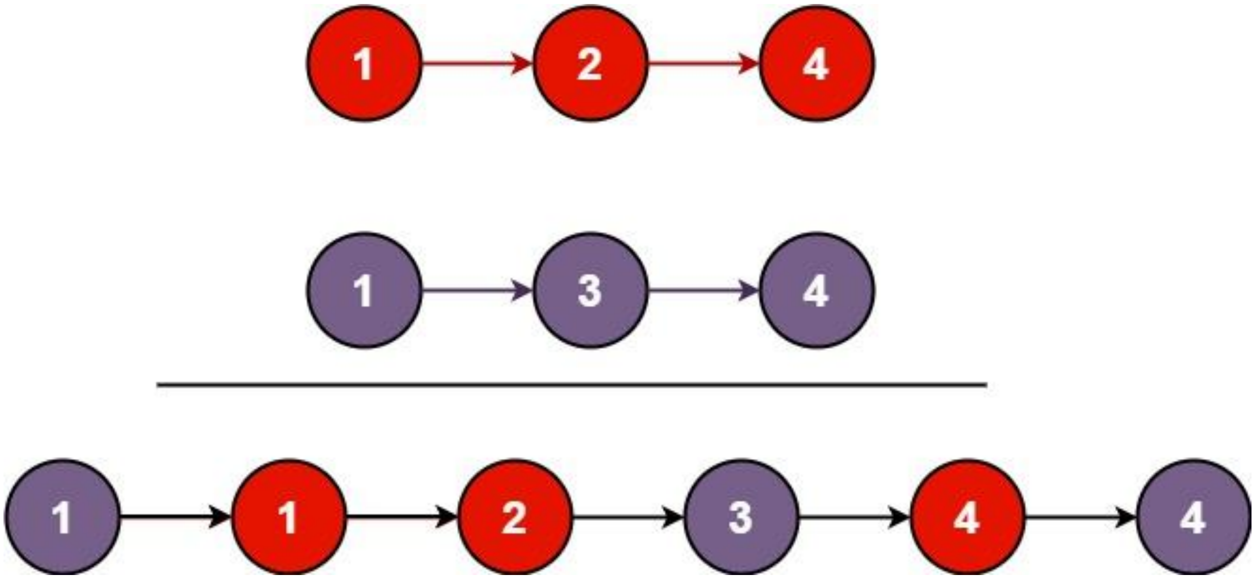
Sorted Lists

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]

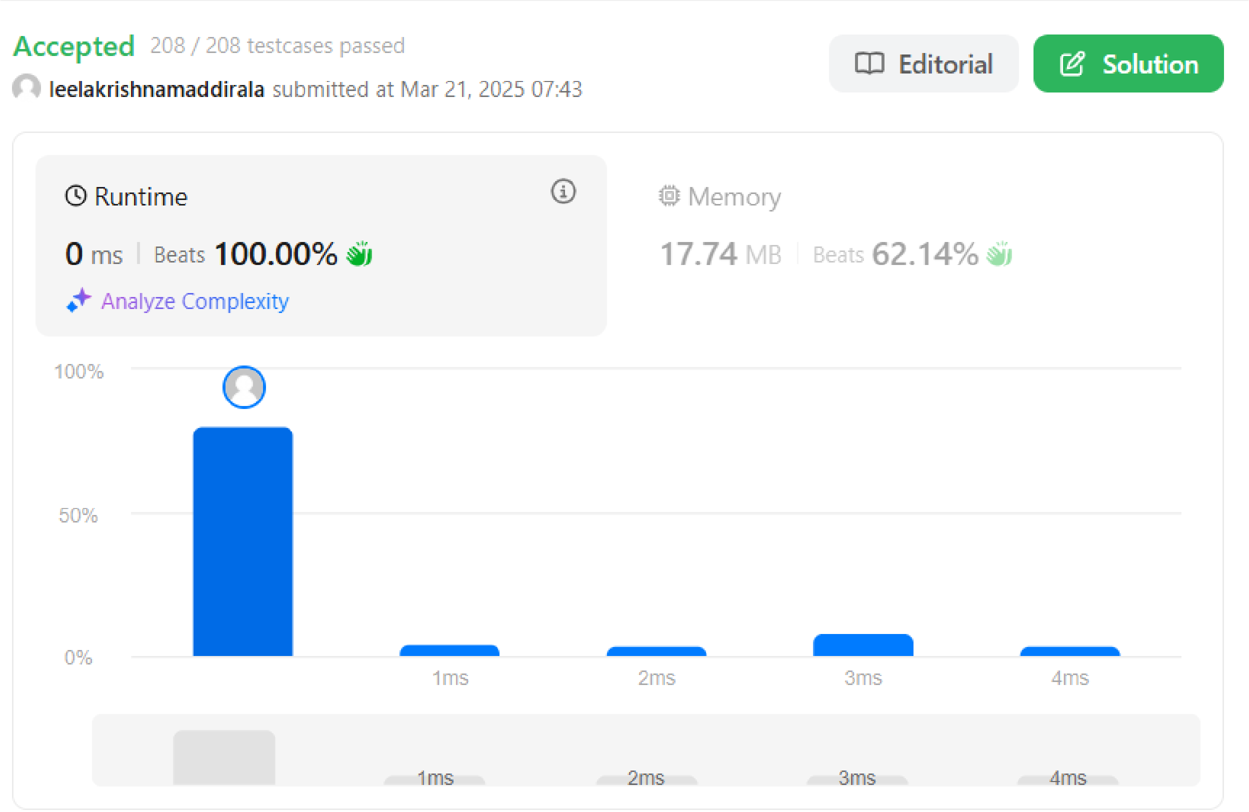
Output: [0]

Constraints:

- The number of nodes in both lists is in the range [0, 50].

- -100 <= Node.val <= 100
- Both list1 and list2 are sorted in **non-decreasing** order.

```
class Solution:
    def mergeTwoLists(self, list1: Optional[ListNode], list2: Optional[ListNode]) -> Optional[ListNode]:
        dummy=ListNode(0)
        current=dummy
        while list1 and list2:
            if list1.val<= list2.val:
                current.next =list1
                list1=list1.next
            else:
                current.next = list2
                list2=list2.next
            current = current.next
        if list1:
            current.next = list1
        if list2:
            current.next = list2
        return dummy.next
```



Accepted 208 / 208 testcases passed

leelakrishnamaddirala submitted at Mar 21, 2025 07:43

Runtime

0 ms | Beats 100.00%

Memory

17.74 MB | Beats 62.14%