Merge Two Sorted Lists

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def mergeTwoLists(self, list1: Optional[ListNode], list2: Optional[ListNode]) → (
```

My logic would be

O(n) approach which compares two lists by the position of the pointers

Compare pointers, and move pointer across until the list is exhausted and append whatever remains

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def mergeTwoLists(self, list1: Optional[ListNode], list2: Optional[ListNode]) → (
        dummy = ListNode(-1)
        current = dummy

        if not list1:
          return list2
        if not list2:
          return list1

        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

            else:
                current.next = list2

        return dummy.next
```

As for the dummy logic, we initialise dummy to be the results linked list

- first assign dummy head to be our current value
- as we progress through the values, we refer to the next number in line
- when all is finished, we return the 2nd number to the head of the dummy list