21. Merge Two Sorted Lists

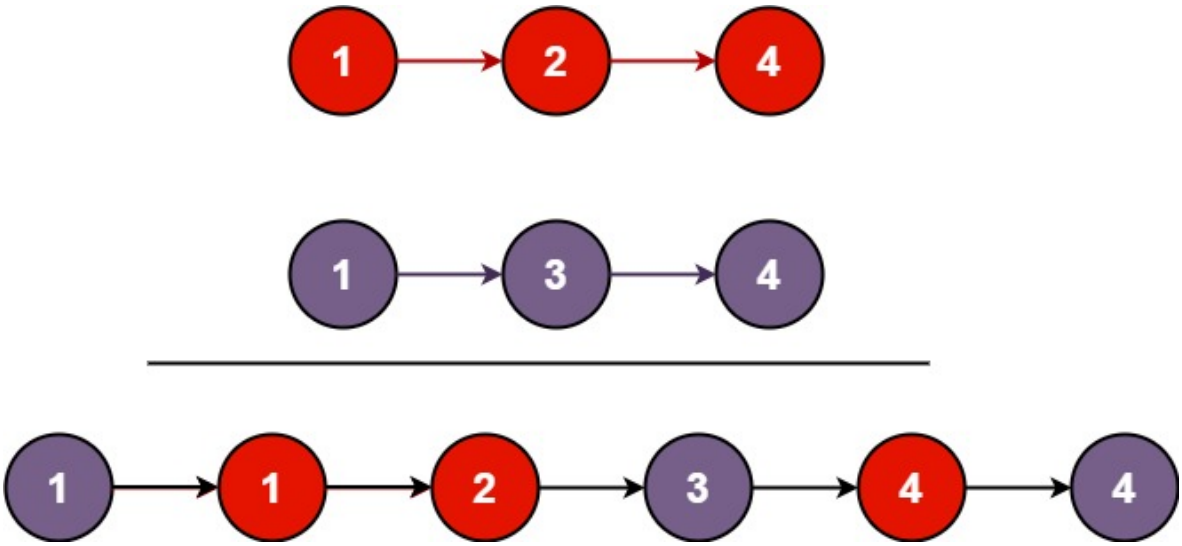
Created	@February 23, 2022 10:21 PM
Difficulty	Easy
LC Url	https://leetcode.com/problems/merge-two-sorted-lists/
Importance	
Tag	Array&Sorting Two pointers
Video	

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

Merge the two lists in a 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.

Solution

```
# 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]) -> Optional[ListNode]:
        dummy = ListNode(-1)
        p = dummy
        p1, p2 = list1, list2

        while p1 and p2:
            if p1.val > p2.val:
                p.next = p2
                p2 = p2.next
            else:
                p.next = p1
                p1 = p1.next
            p = p.next

        if p1:
            p.next = p1

        if p2:
            p.next = p2

        return dummy.next
```

linked list不耗费额外的空间

88. Merge Sorted Array

O(n)的额外的空间

```
def mergeSortedArray(self, A, B):
    i, j = 0, 0
    C = []
    while i < len(A) and j < len(B):
        if A[i] < B[j]:
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1

    while i < len(A):
        C.append(A[i])
        i += 1

    while j < len(B):
        C.append(B[j])
        j += 1

    return C
```