정렬되어 있는 두 연결 리스트를 합쳐라.

Input: l1 = [1,2,4], l2 = [1,3,4]
Output: [1,1,2,3,4,4]

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
```

1.브루트 포스

```
class Solution:
    def mergeTwoLists(self, l1: ListNode, l2: ListNode) -> ListNode:
        result = None
        while l1 or l2:
            if not l1:
                result = ListNode(val = l2.val, next = result)
                l2 = l2.next
            elif not l2:
                result = ListNode(val = l1.val, next = result)
                l1 = l1.next
            elif l1.val == l2.val:
                result = ListNode(val = l1.val, next = result)
                result = ListNode(val = l2.val, next = result)
                l1 = l1.next
                l2 = l2.next
            elif l1.val < l2.val:
                result = ListNode(val = l1.val, next = result)
                l1 = l1.next
            elif l1.val > l2.val:
                result = ListNode(val = l2.val, next = result)
                l2 = l2.next
        rresult = None
        while result:
                rresult = ListNode(val = result.val, next = rresult)
                result = result.next
        return rresult
```

2. 재귀 호출

```
class Solution:
    def mergeTwoLists(self, l1: ListNode, l2: ListNode) -> ListNode:
        if (not l1) or (l2 and l1.val > l2.val):
            l1, l2 = l2, l1
        if l1:
            l1.next = self.mergeTwoLists(l1.next, l2)
        return l1
```

- l1, l2를 비교하여 작은것을 l1에 있게함,
- l1.next를 l2와 다시비교한다.