# Problem 21: Merge Two Sorted Lists

## Problem Information

**Difficulty:** Easy
**Acceptance Rate:** 67.54%
**Paid Only:** No
**Tags:** Linked List, Recursion

## Problem Description

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:**

![](https://assets.leetcode.com/uploads/2020/10/03/merge_ex1.jpg)

**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.

## Code Snippets

**C++:**

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {

}
};
```

**Java:**

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode mergeTwoLists(ListNode list1, ListNode list2) {

}
}
```

**Python3:**

```
# 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]:
```