

21. 合并两个有序链表

地址：[🔗 合并两个有序链表](#)

题目

中文

21. 合并两个有序链表

难度 简单 1791 收藏 分享 切换为英文 接收动态 反馈

将两个升序链表合并为一个新的 **升序** 链表并返回。新链表是通过拼接给定的两个链表的所有节点组成的。

示例 1：

输入: l1 = [1,2,4], l2 = [1,3,4]
输出: [1,1,2,3,4,4]

示例 2：

输入: l1 = [], l2 = []
输出: []

示例 3：

输入: l1 = [], l2 = [0]
输出: [0]

提示：

- 两个链表的节点数目范围是 `[0, 50]`
- `-100 <= Node.val <= 100`
- `l1` 和 `l2` 均按 **非递减顺序** 排列

通过次数 619,598 | 提交次数 934,455

English

21. Merge Two Sorted Lists

难度 简单 1791 收藏 分享 切换为中文 接收动态 反馈

Merge two sorted linked lists and return it as a **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Example 1:

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

Example 2:

Input: l1 = [], l2 = []
Output: []

Example 3:

Input: l1 = [], l2 = [0]
Output: [0]

Constraints:

- The number of nodes in both lists is in the range `[0, 50]`.
- `-100 <= Node.val <= 100`
- Both `l1` and `l2` are sorted in **non-decreasing** order.

审题

- 给定两个链表，都是升序链表，合并成一个升序链表。

思路 1: 递归

★ 分析：

★ 判断 l1 和 l2 头结点哪个更小，然后较小结点的 next 指针指向 其余结点的合并结果。（调用递归）

代码

```
1 // Java
2 // Time : 2021 - 07 - 17
3
4 public ListNode mergeTwoLists(ListNode l1, ListNode l2) {
5     if (l1 == null)          // l1 链表为空, 返回 l2 链表
6         return l2;
7     else if (l2 == null)     // l2 链表为空, 返回 l1 链表
8         return l1;
9     else if (l1.val < l2.val) {          // l1 头节点值 < l2 头节点值
10         l1.next = mergeTwoLists(l1.next, l2); // 将 l1 next 指针指向 剩余结果的并集
11         return l1;                        // 递归返回 l1
12     } else {                             // l2 头节点值 < l1 头节点值
13         l2.next = mergeTwoLists(l1, l2.next); // 将 l2 next 指针指向 剩余结果的并集
14         return l2;                        // 递归返回 l2
15     }
16 }
```