



链滴

# [每日 LeetCode] 21. Merge Two Sorted Lists

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1556109399764>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Description:

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

**Example:**

Input: 1->2->4, 1->3->4

Output: 1->1->2->3->4->4

---

思路：本题要求合并两个有序的链表。同样有迭代和递归的方法。

迭代方法：以 **l1** 链表为主，依次遍历 **l2**，将 **l2** 的节点合并到 **l1** 中。

递归方法：每次返回 **l1** 和 **l2** 中较小的节点。

---

C++代码 (迭代)

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        if (l1 == nullptr)
            return l2;
        if (l2 == nullptr)
            return l1;
        if (l1->val > l2->val)
            swap(l1, l2);
        ListNode* head = l1, *tail = l1;
        l1 = l1->next;
        while(l1 && l2){
            if (l1->val < l2->val){
                tail->next = l1;
                tail = l1;
                l1 = l1->next;
            }
            else{
                tail->next = l2;
                tail = l2;
                l2 = l2->next;
            }
        }
        if (l1 != nullptr)
```

```
        tail->next = l1;
    if (l2 != nullptr)
        tail->next = l2;
    return head;
}
};
```

运行时间: 8ms

运行内存: 9M

---

C++代码 (递归)

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        ListNode* head = NULL;
        if(l1 == NULL)
        {
            head = l2;
        }
        else if(l2 == NULL)
        {
            head = l1;
        }
        else
        {
            if(l1->val < l2->val)
            {
                head = l1;
                head->next = mergeTwoLists(l1->next, l2);
            }
            else
            {
                head = l2;
                head->next = mergeTwoLists(l1, l2->next);
            }
        }
        return head;
    }
};
```

运行时间: 12ms

运行内存: 9M

---