

剑指25 合并两个排序的链表

核心思路类似，依然分为递归和迭代：

法一：迭代。引入伪头节点，双指针交替遍历串联起链表。（时间空间利用最佳）

法二：递归

我的初始办法：插入

我的法一实现：

```
1 class Solution {
2 public:
3     ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
4         ListNode faker = {0, NULL};
5         ListNode* p = &faker;
6         while(l1 != NULL && l2 != NULL) {
7             if (l1->val < l2->val) {
8                 p->next = l1;
9                 l1 = l1->next;
10            }
11            else {
12                p->next = l2;
13                l2 = l2->next;
14            }
15            p = p->next;
16        }
17        p->next = (l1 == NULL) ? l2 : l1;
18        return faker.next;
19    }
20 };
```

官方的法二实现：

```
1 class Solution {
2 public:
3     ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
4         if (l1 == nullptr) {
5             return l2;
6         } else if (l2 == nullptr) {
7             return l1;
8         } else if (l1->val < l2->val) {
9             l1->next = mergeTwoLists(l1->next, l2);
10            return l1;
11        } else {
12            l2->next = mergeTwoLists(l1, l2->next);
13            return l2;
14        }
15    }
16 };
```

```
14         }
15     }
16 };
17 // 递归的关键状态：
18 // 结果 = 当前l1, l2中小的那个串联上后面假设已经排好序的链表
19 // 也就是说，两个链表头部值较小的一个节点与剩下元素的 merge 操作结果合并。
```