知识小料

「电计 2203 班」周常规知识整理共享

21

日期：2024-3-30

学科：数据结构与算法

阅读以下算法，回答下列问题。

```
1  LinkList mynote(LinkList L)
2  { //L是不带头结点的单链表头指针
3      if(L&&L->next)
4      {
5          q=L; L=L->next; p=L;
6  S1:     while(p->next) p=p->next;
7  S2:     p->next=q; q->next=NULL;
8      }
9      return L;
10 }
```

1. 说明语句 S1 的功能；
2. 说明语句组 S2 的功能；
3. 设链表表示的线性表为 $(a_1, a_2, \ldots, a_n)$，写出算法执行后的返回值所表示的线性表。

这是一个典型的链表操作算法。~~很容易看出它根本无法编译。~~

单链表 L 不带表头结点，因此表头位置实实在在存储了元素；p, q 是两个指针。我们用赛跑的情景描述这个算法，逐行阅读它：

- 第 3 行，如果表中至少有 2 个元素，就可以进入条件。（不是循环）
- 第 5 行，p 跑在 q 的「前面」（即下一个位置），q 在「起点」。
- 第 6 行 (S1)，我们让 p 不断地探路：p 会「向前跑」，直到到达「终点」。
- 第 7 行 (S2)，我们把 q 所指的位置成为 p 的下一个元素，并让 q 的下一个为空。这也就意味着，程序把 q 所指的「起点」位置挪到了「终点」的下一个位置，原来的「起点」变成了新的「终点」，如图 1 所示。
- 第 9 行，一轮过程结束后，算法结束，返回整个链表。

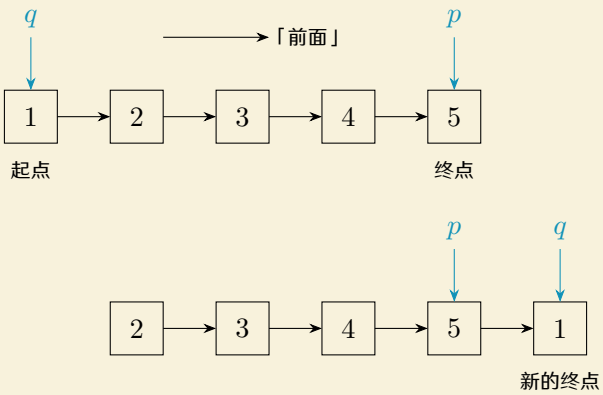

图 1: 执行第 7 行前后的图解（`p->next=q; q->next=NULL;`）

把刚才的思考过程转化成书面语：「前面」= 下一个结点，「起点」= 第一个结点，「终点」= 最后一个结点。返回的线性表在图中有所体现，就是 $(a_2, a_3, \ldots, a_n, a_1)$。这样我们就得到以下答案。

【结论】

1. 查询链表的尾节点（遍历链表）
2. 将第一个结点链接到链表的尾部，作为新的尾结点
3. 返回的线性表为 $(a_2, \ldots, a_n, a_1)$

【点评】本题是链表算法的阅读理解问题，对链表元素的移动和拼接（插入）作出了考察。理解「后继结点」的赋值操作的真正含义是解决此题的关键。