

本小节内容

题目解读

解题设计

1 题目解读

2019 年(单链表)

41. (13 分) 设线性表 $L = (a_1, a_2, a_3, \dots, a_{n-2}, a_{n-1}, a_n)$ 采用带头结点的单链表保存, 链表中的结点定义如下:

```
typedef struct node {  
    int data;  
    struct node* next;  
} NODE;
```

请设计一个空间复杂度为 $O(1)$ 且时间上尽可能高效的算法, 重新排列 L 中的各结点, 得到线性表 $L' = (a_1, a_n, a_2, a_{n-1}, a_3, a_{n-2}, \dots)$ 。要求:

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。
- (3) 说明你所设计的算法的时间复杂度。

解读:

首先空间复杂度是 $O(1)$, 我们不能申请额外的空间, 然后找到链表的中间结点, 前面一半是链表 L , 将链表的后半部分给一个新的头结点 L_2 , 然后将链表 L_2 进行原地逆置, 然后再将 L 和 L_2 链表进行合并

2 解题设计

第一阶段, 如何找到链表的中间结点, 可能不少同学的思考是首先遍历一次链表, 长度假如是 20, 再次遍历走到第 10 个, 这样的缺点是遍历了两次链表。这里我们可以练习两个指针同步向后遍历的方法。我们定义两个指针 $pcur$, $ppre$, 我们让 $pcur$ 指针每次走两步, $ppre$ 指针每次走一步, 这样当 $pcur$ 指针走到最后, 那么 $ppre$ 指针刚好在中间。这里理解一定要通过自己动手画图去理解, 会非常清晰。担心链表结点总数目是奇数个, 还是偶数个这个问题, 也可以通过画图去分析。

注意, 由于 $pcur$ 每次循环是走两步的, 因此每走一步都注意判断是否为 NULL。

第二阶段, 后半链表我们设置为了 L_2 , 如何让 L_2 原地逆置? 首先我们需要判断链表是否为空, 如果为空, 就返回, 如果只有 1 个结点, 也不需要逆置, 直接返回。

第一步: 链表原地逆置, 我们需要使用 3 个指针, 假如分别是 r , s , t , 它们分别指向链表的 1, 2, 3, 也就是前三个结点。

第二步: 我们让 $s \rightarrow next = r$, 这样 2 号结点就指向了 1 号结点, 完成了逆置。

第三步: 这时, $r = s$, $s = t$, $t = t \rightarrow next$, 通过这个操作, r, s, t 分别指向了链表的 2, 3, 4 结点,

这时回到第二步，循环往复，当 t 为 $NULL$ 时，结束循环。

第四步：循环结束时， t 为 $NULL$ ，这时 s 是最后一个结点， r 是倒数第第二个结点，需要再次执行一下 $s \rightarrow next = r$ 。

第五步：最后需要 $L2 \rightarrow next \rightarrow next = NULL$ ；因为原有链表的头结点变成链表最后一个结点，最后一个结点的 $next$ 需要为 $NULL$ ，这时让 $L2$ 指向 s ，因为 s 是原链表最后一个结点，完成了逆置后，就是第一个结点，因此链表头结点 $L2$ 指向 s 。

以上设计未写过的同学一定要通过画图来理解，我们视频也是通过画图讲解设计。

第三阶段：将 L 与 $L2$ 链表合并，合并时轮流放入一个结点。因为空间复杂度是 $O(1)$ ，因此我们不申请新空间，但是我们依然需要 3 个指针 ($pcur, p, q$)，合并后的新链表我们让 $pcur$ 指针始终指向新链表尾部，初始化为 $pcur = L \rightarrow next$ ，使用 p 指针始终指向链表 L 待放入的结点，初始化值为 $p = L \rightarrow next$ ， q 指针始终指向链表 $L2$ 待放入的结点，初始化值为 $q = L2 \rightarrow next$ 。因为链表 L 的第一个结点不动，所以 $p = p \rightarrow next$ 。

开启循环 $while(p \neq NULL \&\& q \neq NULL)$ ，首先将 $pcur \rightarrow next = q$ ，然后 $q = q \rightarrow next$ 和 $pcur = pcur \rightarrow next$ ，接着 $pcur \rightarrow next = p$ ，然后 $p = p \rightarrow next$ 和 $pcur = pcur \rightarrow next$ ，直到循环结束。循环结束后，有可能 L 还剩余一个结点，也可能 $L2$ 剩余一个结点，但是只会有一个剩余的有结点，因此我们判断 p 不为 $NULL$ ，把 p 放入，如果 q 不为 $NULL$ ，把 q 放入即可。

以上设计未写过的同学一定要通过画图来理解，我们视频也是通过画图讲解设计。