

第二次作业

120L022314 瞿久尧

1. 两个链表分别为 link1 和 link2, 设 link1 里的元素为 x, link2 里的元素为 y, 定义指针变量 int *temp, 用于储存地址。

方法: 将 link1 中的 x 元素逐个插入到 link2 中

从头结点开始, x 和 y 进行比较

若 $x > y$, 则令指针变量 $temp = y \rightarrow next$, 用于储存指针, 然后 $y = y \rightarrow next$, 若仍有 $x > y$, 重复上述操作, 直到 $x = y$ 或 $x < y$;

若 $x = y$, 则删除 x 结点, 即: 使 x 的前驱的指针直接指向 x 的后继, 得到 $x \rightarrow next$;

若 $x < y$, $*temp = x$, $x \rightarrow next = y$, 即 x 的前驱变成 y 的前驱, x 的后继变成 y。(即: 将 x 放入 link2 的两个结点之间, 且满足大小关系递增排序)

反复进行如上操作, 若先 $x \rightarrow next == NULL$, 两个链表已经以递增的顺序合并为一个链表 link2;

若先 $y \rightarrow next == NULL$, 则令 $y \rightarrow next = x$, 使剩余的 x 接在 link2 的末尾, 同样合并为一个链表。

2. 两个链表分别为 link1 和 link2, 设 link1 里的元素为 x, link2 里的元素为 y, 定义指针变量 int *temp, 用于储存地址。

方法: 逐个比较大小, 确定出重复的元素, 并且使之串联(前驱与后继的关系), 同时可以将不重复的部分切除。

从头结点开始, 在 link1 中, 先令 $temp = head \rightarrow next$, 然后依次比较 x 和 y

若 $x > y$, 则 $y = y \rightarrow next$;

若 $x < y$, 则 $x = x \rightarrow next$;

若 $x = y$, 则令 $*temp = x$, $temp = x \rightarrow next$, $x = x \rightarrow next$, 继续进行比较;

直到比较完成, 以 NULL 为终点, link1 是以 head 为头结点的链表 A 和 B 的交集, 并且以递增排列

3. 定义三个指针 int *temp1, *temp2, *temp3, 设链表中的元素为 x, 从头结点开始

方法: 若以左为头, 右为尾, 此方法便是将单链表的方向改变, 以右为头, 左为尾, 实现翻转的目的, 且仍为原来的链表空间

$temp1 = x$; //将第一个结点的地址存在指针 1 中

$temp2 = x \rightarrow next$; //将第一个结点的后继的地址存在指针 2 中

$x = x \rightarrow next$; //x 变成第二个结点

$*temp2 = NULL$; //即, 将头结点变为尾结点

$temp3 = temp1$; //将第一个结点的地址存在指针 3 中

$temp1 = x$; //将第二个结点的地址存在指针 1 中

$temp2 = x \rightarrow next$; //将第二个结点的后继的地址存在指针 2 中

$x = x \rightarrow next$; //x 变成第三个结点

$*temp2 = temp3$ //把第一个结点变成第二个结点的后继, 实现反向

..... //重复以上高亮操作, 直到最后一个

其中可以用 `count`，初始化为 0，用来计数，可以根据链表的长度来控制计数多少次后停止

4. 设链表中的元素为 `a`，定义指针变量 `int *temp`

从头结点开始，比较 `a` 和 `x`

方法：用指针确定 `a1 < x` 的位置，同时推至 `a2 > y` 的位置，将 `a1->next=a2`，以实现中间部分链表被切除的目的

若 `x > a`，令 `temp=a->next`;

然后 `a=a->next`，继续进行比较，重复操作；

直到 `a >= x`，此时 `*temp` 固定，然后将 `a` 与 `y` 比较；

若 `a <= y`，`a=a->next`，继续比较，重复操作；

直到 `a > y`，令 `*temp=a`，即可删除链表中大于等于 `x`，且小于等于 `y` 的数

周三作业：

转换八进制的时间复杂度 $O(\log_8 n)$

由于基本操作的时间复杂度为 $O(1)$ ，操作进行 $\log_8 n$ 次