

1. 总结一下线性表的特点。

①集合中存在唯一一个“第一元素”和“最后一元素”。②除第一元素外，其他每个元素有且仅有一个直接前驱。③除最后一元素外，其他每个元素有且仅有一个直接后继。

2. 顺序表插入算法 $\text{Insert}(\text{Sqlist} * L, \text{int } i, \text{ElemType } e)$ 实现过程中，如果有 n 个元素，在 i 位置插入，则需要向后移动多少个元素？第一个移动的元素是那个位置？最后一个移动的元素是那个位置？

移动： $n - (i - 1)$ 次。 第一个移动的是 n 位置。 最后一个： i 位置。

3. $\text{Delete}(\text{Sqlist} * L, \text{int } i)$ 实现过程中，有 n 个元素，删除 i 位置元素，需向前移动多少个元素？第一个移动元素是那个位置，最后移动元素是那个位置？

移动 $n - i$ 个。 第一个移动： $i + 1$ 个位置。 最后移动： n 个位置。

4. 总结顺序表优缺点。

优：逻辑相邻，物理相邻；可随机存取任一元素；存取空间使用紧凑。

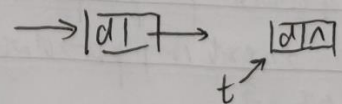
缺：插入、删除操作要大量元素移动；预先分配空间需按最大空间分配；利用不充分；表容量难以扩充。

5. 构造先进先出单链表（也称为尾插法），新插入的元素在表头还是表尾？每次插入新结点时的操作目标是哪一步？

①读入数据并申请一个新结点空间。

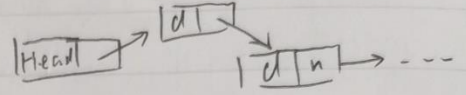
②将尾指针的 next 指向新结点。

③尾指针指向新结点。



6. 构造后进先出单链表(头插法) 新插入在表头 表尾? 新插入结点步骤

- ① 读入数据并申请新结点空间.
- ② 新结点指向头指针的 next.
- ③ 头指针指向新结点.



7. 单链表中第 i 位置插入新元素 e. 总结思路.

- ① 为 e 申请空间
- ② 遍历找到 i 位置.
- ③ i-1 的 next 指向新元素 e.
- ④ e 的 next 指向 i.

8. 中删除第 i 个结点. 总结思路.

- ① 遍历找到 i-1 位置.
- ② i-1 的 next 指向 i+1.
- ③ 释放 i 位置空间.

9. 在单链表中删除值为 e 的结点. 总结思路.

- ① 遍历找到值为 e 的结点位置. 并用指针 p 记录上一个结点位置.
- ② p 指向值为 e 结点的 next.
- ③ 释放值为 e 结点空间.

10. 有序单链表 pa, pb, pc. 三个指针作用? 若 $pa \rightarrow data < pb \rightarrow data$ 该怎么办? (代码)

pa: a 链元素.

pb: b 链元素.

pc: 新合并的链表的尾元素.

```

if (pa->data < pb->data) else
{ pc->next = pa;          { pc->next = pb;
  pc = pa;                  pc = pb;
  pa = pa->next;            pb = pb->next;
}                             }
    
```


11. 相比较单链表, 单循环链表的区别是什么? 优点?

单循环链表是单链表的变形, 单循环链表最后一个结点的next指向头结点。

优点 ①不用预先分配存储空间。

②知道任一结点位置可知所有结点位置。

③一种动态结构。

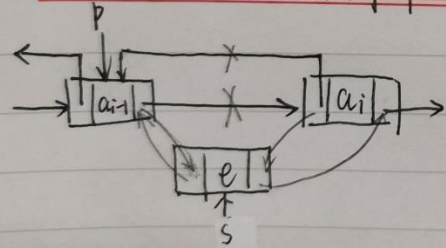
12. 双向链表结点结构部分, 分别存储着什么信息?

back	data	next
左链指针	数据	右链指针

← 前趋

后继 →

13. 如下图所示双向链表中, 在p指针指向的结点后面插入s指针指向的结点, 给出操作语句。



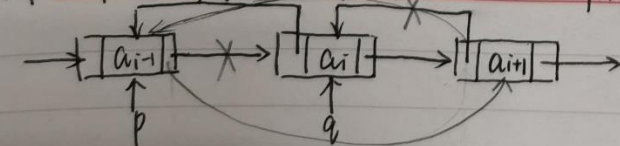
$s \rightarrow \text{next} = p \rightarrow \text{next};$

$s \rightarrow \text{prior} = p;$

$\Delta p \rightarrow \text{next} \rightarrow \text{prior} = s;$

$p \rightarrow \text{next} = s$

14. 如下图所示双向循环链表中, 删除q指针指向的结点, 给出语句。



$p \rightarrow \text{next} = q \rightarrow \text{next};$

$q \rightarrow \text{next} \rightarrow \text{prior} = p;$