线性表

目 录

Contents

- 线性表
  - 线性存储 —— 顺序表
  - 链式存储
    - 单链表
    - 双链表
    - 循环链表
    - （指针实现）
    - 静态链表（借助数组实现）

定义和基本操作

线性表：线性表是具有相同数据类型的n（n>=0）个数据元素的有限序列。一种典型的线性结构。除第一个元素外，每个元素有且仅有一个直接前驱。除最后一个元素外，每个元素有且仅有一个直接后继。

一维数组可以静态分配，也可以是动态分配。

静态分配:数组的大小和空间事先已固定，一旦空间占满，再加入新的数据将产生溢出，会导致程序崩溃

动态分配：储存数据的空间实在程序执行过程中通过动态存储分配语句分配，一旦空间占满，可以另外开辟一块更大的储存空间，用来替换原来的存储空间。

定义和基本操作

线性表的特点：

1 表中元素的个数有限

2表中元素具有相同逻辑上的顺序性，在序列中各元素有先后次序。

3 表中元素都是数据元素，每一个元素都是单个元素

4 表中元素的数据类型相同。这意味着每一个元素占有相同大小的存储空间

5 表中元素具有抽象性。即仅讨论元素间的逻辑关系，不考虑元素究竟表示什么内容

定义和基本操作

线性表(Linear List)：是由n(n≥0)个数据元素(结点)a1，a2，…，an组成的有限序列。该序列中的所有结点具有相同的数据类型。其中数据元素的个数n称为线性表的长度。

当n=0时，称为空表。

当n>0时，将非空的线性表记作：(a1，a2，…an)

a1称为线性表的第一个(首)结点，an称为线性表的最后一个(尾)结点。

定义和基本操作

线性结构是最常用、最简单的一种数据结构。而线性表是一种典型的线性结构。其基本特点是线性表中的数据元素是有序且是有限的。在这种结构中：

① 存在一个唯一的被称为“第一个”的数据元素；

② 存在一个唯一的被称为“最后一个”的数据元素；

③ 除第一个元素外，每个元素均有唯一一个直接前驱；

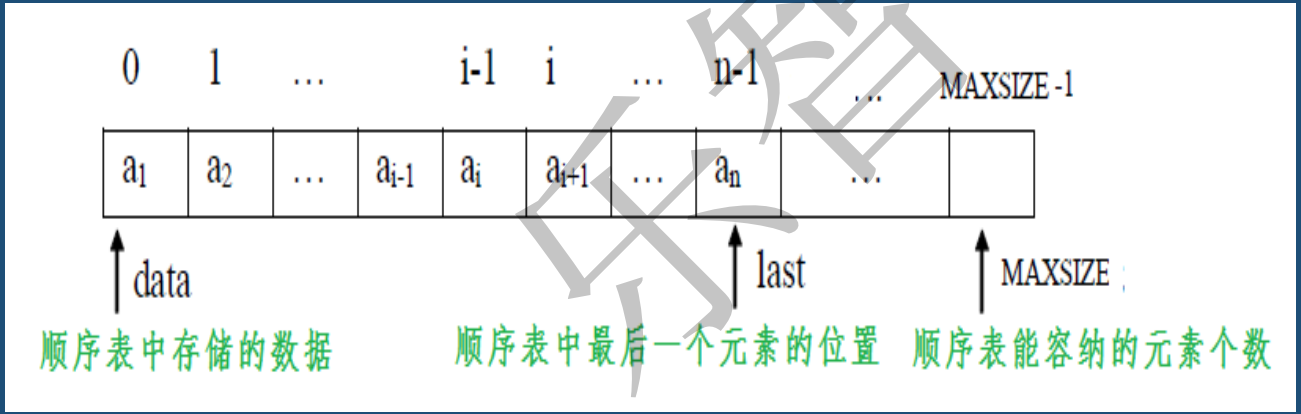
④ 除最后一个元素外，每个元素均有唯一一个直接后继。

线性表的顺序表示（顺序表）

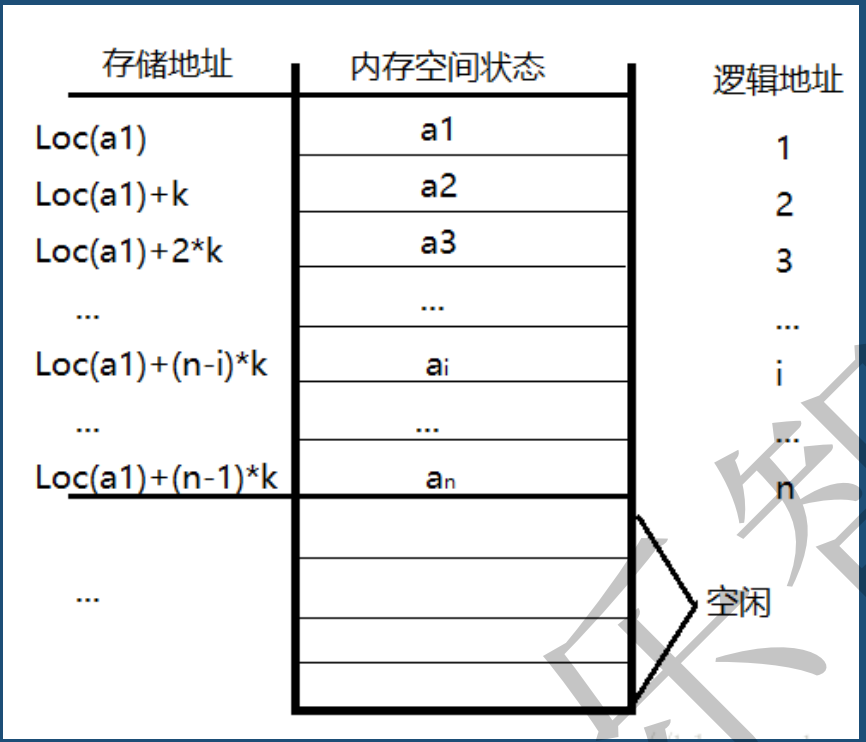
线性表的顺序存储又称为**顺序表**，它是用一组地址连续的存储单元，依次存储线性表中的数据元素，从而使得**逻辑上相邻的两个元素在物理位置上也相邻**。

特点：表中元素的逻辑顺序与其物理顺序相同。

顺序表最主要的特点是**随机访问**，即通过首地址和元素序号可以在**O(1)**的时间内找到指定的元素。顺序表的**存储密度高，每个结点只存储数据元素**。



顺序表的基本操作（插入操作）



在顺序表的第i个位置插入元素e：

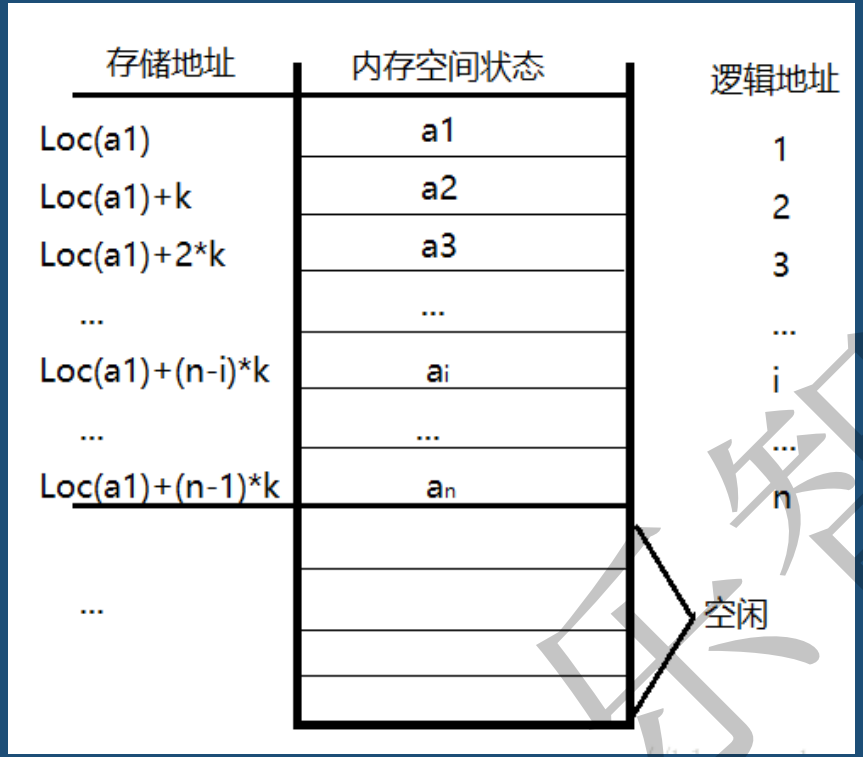
1.首先将顺序表第i个位置的元素依次向后移动一个位置。

2.然后将元素e插入第i个位置。

注意：移动元素要从后往前移动元素，即：先移动最后一个元素，再移动倒数第二个元素，依次类推；插入元素之前要判断插入的位置是否合法，顺序表是否已满，在插入元素之后要将表长L->length++

顺序表的插入操作时间复杂度为O(n)

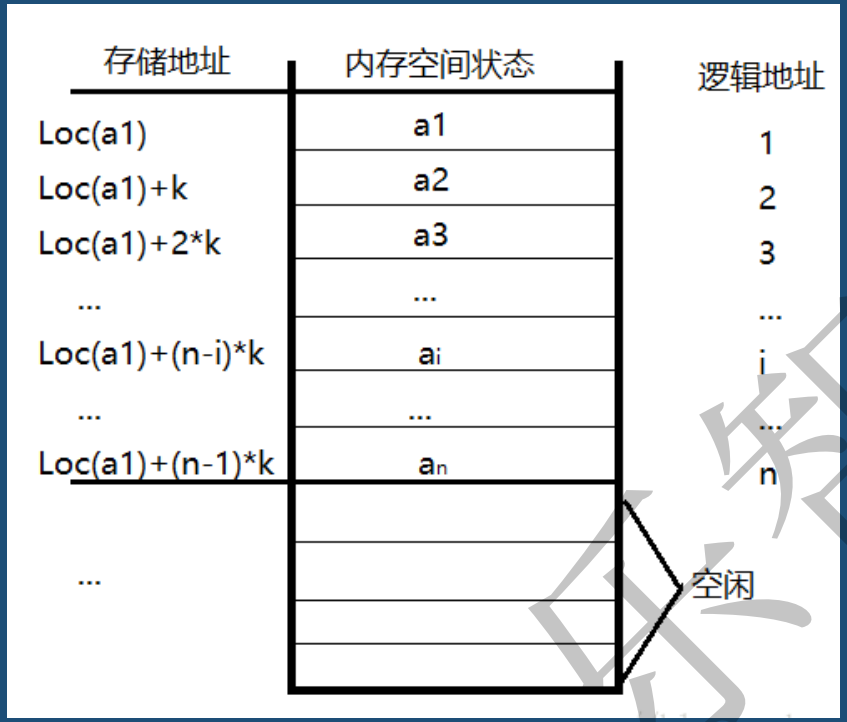
顺序表的基本操作（删除操作）



删除表中的第i个元素e：删除数据表中的第i个元素，需要将表中第i个元素之后的元素依次向前移动一位，将前面的元素覆盖掉。移动元素时要想将第i+1个元素移动到第i个位置，在将第i+2个元素移动i+1的位置，直到将最后一个元素移动到它的前一个位置，进行删除操作之前要判断顺序表是否为空，删除元素之后，将表长**L->length--**;

时间复杂度为O(n)

顺序表的基本操作（按值查找）



在顺序表L中查找元素值等于e的元素的位置序号，并返回其序号

时间复杂度为O(n)

线性表的顺序表示（真题检测）

1.顺序表和一维数组一样，都可以按下标进行随机访问　　（ ）

2.在n个节点的顺序表中，算法的时间复杂度是O(1)的操作是　（ ）

A.访问第i个节点(1≤ i ≤n)和求第i个节点的直接前驱(1≤ i ≤n)
B.在第i个节点后插入一个新节点(1≤ i ≤n)
C.删除第i个节点(1≤ i ≤n)
D.将n个节点从小到大排序

答案：　对　　A

线性表的顺序表示（真题检测）

1.下述哪一条是顺序存储结构的优点（ ）
A.存储密度大 B.插入运算方便 C.删除运算方便
D.方便地运用于各种逻辑结构的存储方式

2.一个顺序表所占用的存储空间大小与（ ）无关
A.表的长度 B.元素的存放顺序 C.元素的类型 D.元素中各字段的类型

3.在一个长度为n的顺序表中删除第i个元素（1<=i<=n）时，需向前移动（ ）个元素
A. n B.i-1 C.n-i D.n-i+1

答案： A B C

线性表的链式表示（单链表）

链表是动态分配存储空间的链式存储结构。

线性表的链式存储又称为单链表。它是指通过一组任意的存储单元来存储线性表中的数据元素。通过指针来建立各数据元素之间的线性关系。

特点：表中元素散列的分布在存储空间中。

单链表的结点结构

通常用头指针来识别一个单链表，头指针指向NULL时，表示一个空链表

Head称为整个链表的头结点，头结点中存放一个地址，该地址指向一个元素，头结点一般不存放具体数据，只是存放第一个结点的地址。

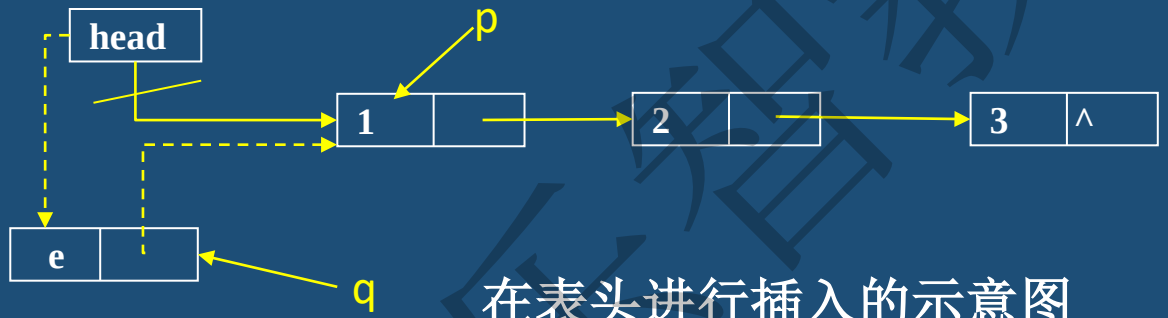
单链表的基本操作（插入操作）

在链表的表头插入：

(1)创建一个新的结点q。

(2)将此结点的数据域赋值为e，并将它的next指针指向第一个结点，即p。

(3)将L修改为指向新的结点q。



在表头进行插入的示意图

```
q->next=p;
L->next=q;
```

时间复杂度为O(1)

单链表的基本操作（插入操作）

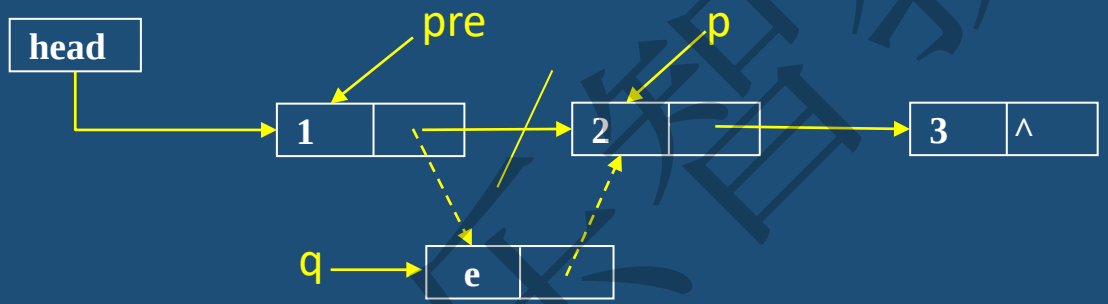
在链表的中间插入：

(1)创建一个新的结点q。

(2)将此结点的数据域赋值为e，并将它的next指针指向p。

(3)查找到p的前驱结点pre。

(4)将pre的next指针指向新创建的结点q。



q->next=pre->next;

pre->next=q;

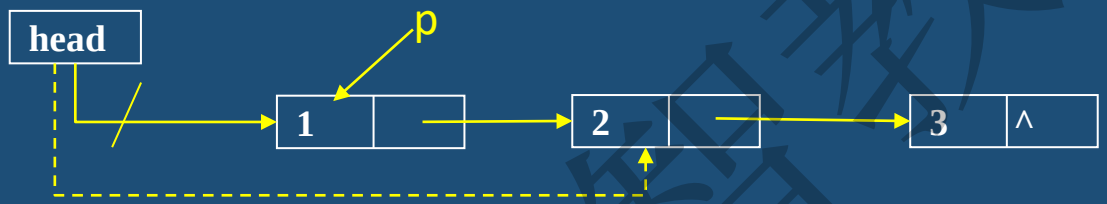
时间复杂度为O(1)

单链表的基本操作（删除操作）

p是链表中的第一个结点：

(1)将L指向p->next。

(2)释放p。




```
L=p->next;
free(p);
```

时间复杂度为O(1)

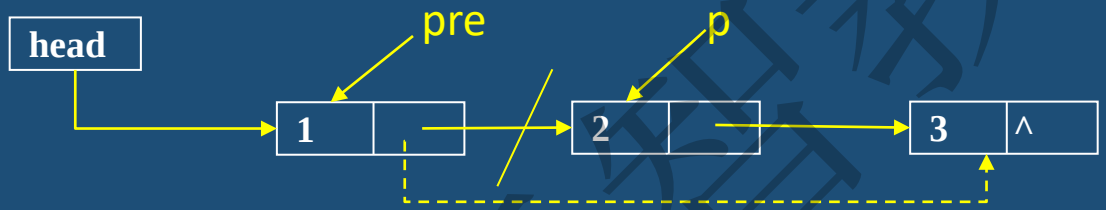
单链表的基本操作（插入操作）

p是链表中的其他结点：

(1)找到p的前驱结点pre。

(2)将pre->next指向p->next。

(3)释放p。

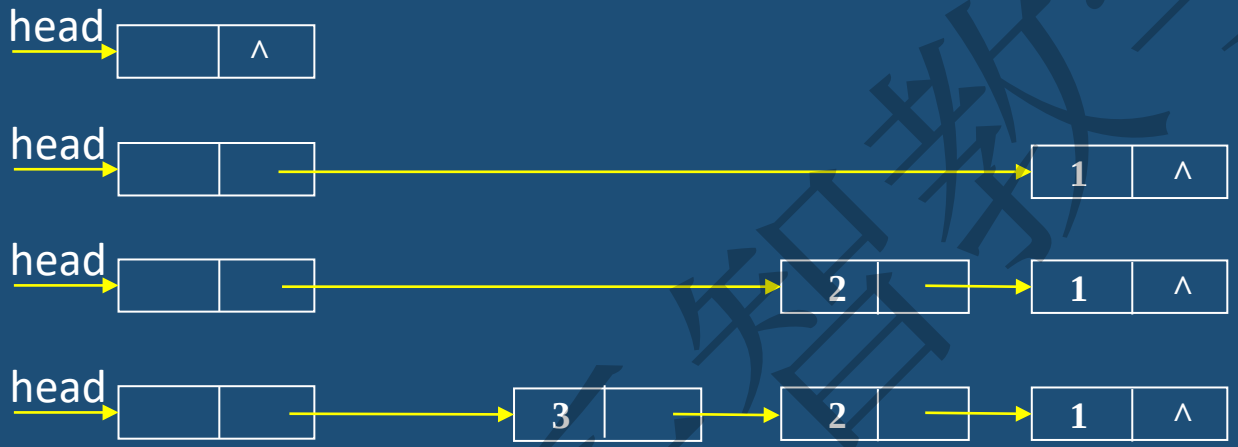



```
pre->next=p->next;
free(p);
```

时间复杂度为O(1)

单链表的基本操作（创建操作）

单链表的创建方法有两种：头插法和尾插法。
头插法是将新增结点插入第一个结点之前，示意图如下：



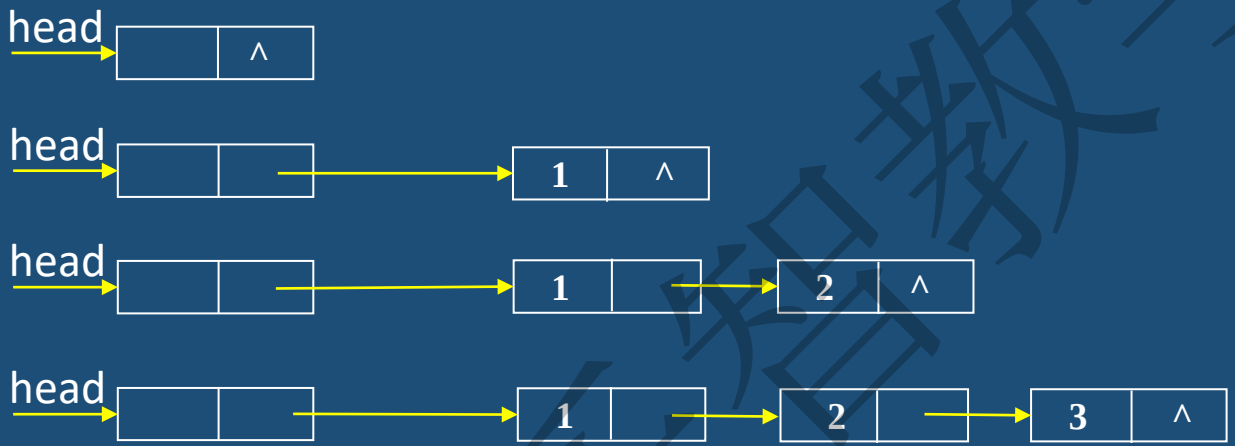
p每次要加入的结点
p->next=L->next;
L->next=p;

时间复杂度为O(n)

单链表的基本操作（创建操作）

单链表的创建方法有两种：头插法和尾插法。

尾插法是将新增结点插入最后一个结点之后，示意图如下：



p是新加入的结点

tail为始终指向表尾结点

```
tail->next=p;
tail=p;
```

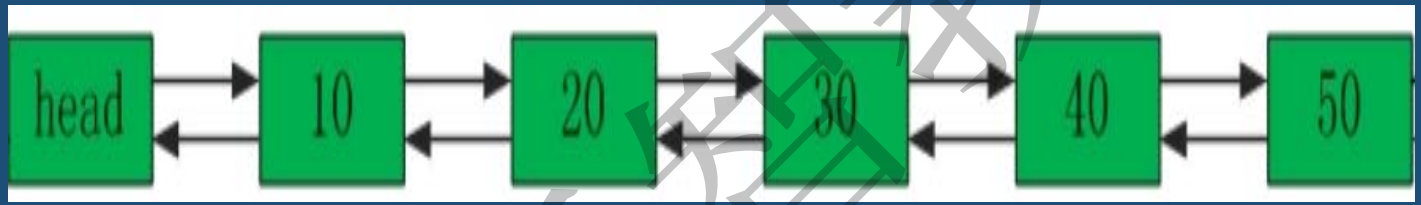
时间复杂度为O(n)

单链表

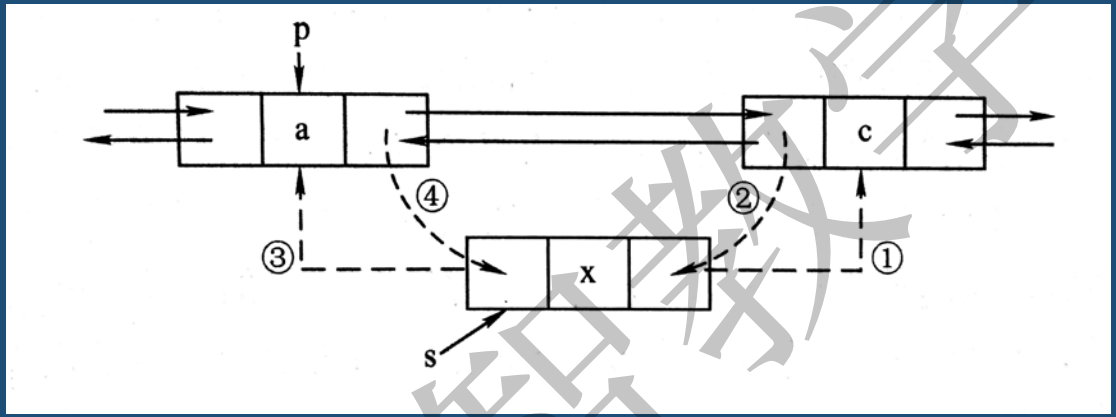
单链表按序号查找结点值的时间复杂度为O(n)
单链表按值找表结点的时间复杂度为O(n)
求表长的时间复杂度为O(n)

线性表的链式表示（双链表）

双向链表(双链表)是链表的一种。和单链表一样，双链表也是由节点组成，它的每个数据结点中都有两个指针，分别指向直接后继(*next)和直接前驱(*prior)。所以，从双向链表中的任意一个结点开始，都可以很方便地访问它的前驱结点和后继结点。

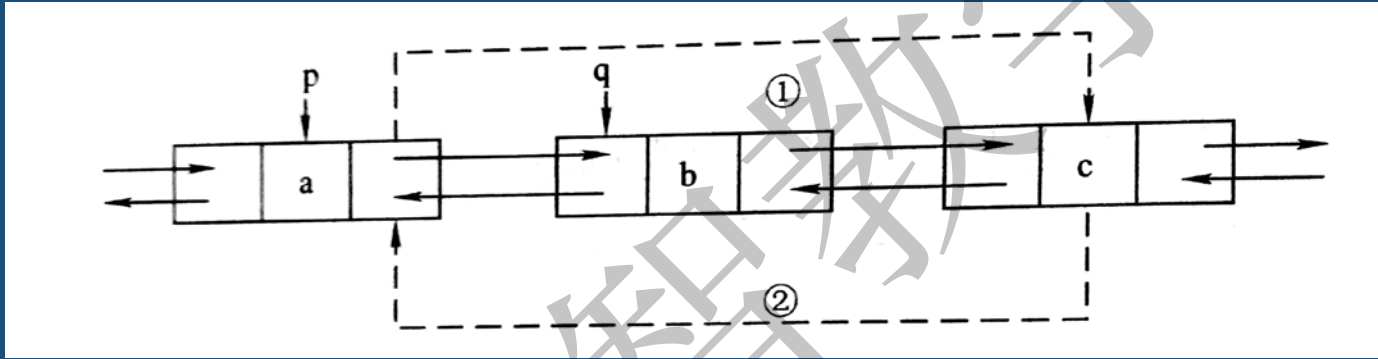


双链表的基本操作（插入操作）




```
s->next=p->next;
p->next->prior=s;
s->prior=p;
p->next=s;
```

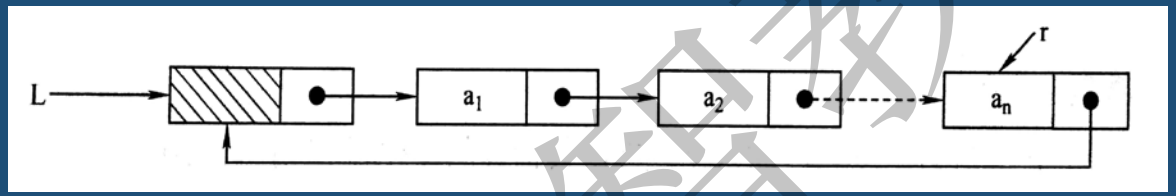
双链表的基本操作（删除操作）

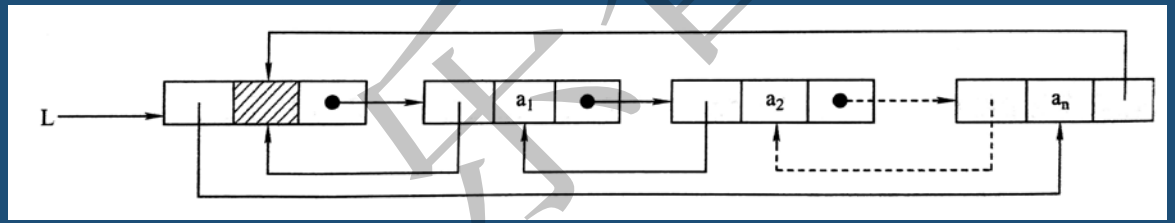



```
p->next=q->next;
q->next->prior=p;
free(q)
```

线性表的链式表示（循环链表）

循环单链表和单链表的区别在于，表中最后一个节点的指针不是NULL，而改为指向头结点，从而整个链表形成一个环。不同于循环单链表，循环双链表中的头节点的prior指针还要指向表尾节点。

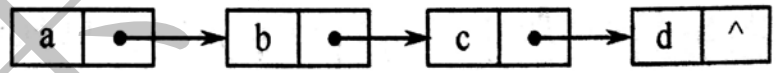




线性表的链式表示（静态链表）

静态链表借助数组来描述线性表的链式存储结构。节点也有数据域data和指针域next，这里的指针是节点的相对地址（数组下标），又称为游标。

0		2
1	b	6
2	a	1
3	d	−1
4		
5		
6	c	3



静态链表以next=-1作为结束的标志，静态链表在做插入、删除操作时不需要移动元素，只需要修改指针。

顺序表和链表的比较

性能类别	具体项目	顺序存储	链式存储
空间性能	存储密度	=1，更优	<1
	容量分配	事先确定	动态改变，更优
时间性能	查找运算	O(n/2)	O(n/2)
	读运算	O(1)，更优	O([n+1]/2),最好情况为1，最坏情况为n
	插入运算	O(n/2),最好情况为0，最坏情况为n	O(1)，更优
	删除运算	O([n-1]/2)	O(1)，更优

顺序表和链表的比较

1.顺序表存储（典型的数组）

原理:顺序表存储是将数据元素放到一块连续的内存存储空间,相邻数据元素的存放地址也相邻(逻辑与物理统一)

优点：（1）空间利用率高。（局部性原理，连续存放，命中率高）

（2）存取速度高效，通过下标直接存储。

缺点：（1）插入和删除比较慢，比如:插入或者删除一个元素时，整个表需要遍历移动元素来重新排一次顺序。

（2）不可以增长长度，有空间限制,当需要存取的元素个数可能多于顺序表的元素个数时会出现“溢出”问题。当元素个数远少于预先分配的空间时,空间浪费巨大。

时间性能：查找O(1) 插入和删除O(n)

顺序表和链表的比较

2.链表存储

原理:链表存储是在程序运行过程中动态的分配空间,只要存储器还有空间,就不会发生存储溢出问题,相邻数据元素可随意存放，但所占存储空间分两部分,一部分存放结点值，另一部分存放表示结点关系间的指针。

优点：（1）存取某个元素速度慢。

（2）插入和删除速度快，保留原有的物理顺序，比如：插入或者删除一个元素时，只需要改变指针即可。

（3）没有空间限制，存储元素的个数无上限，基本只与内存空间大小有关。

顺序表和链表的比较

缺点：（1）占用额外的空间以存储指针（浪费空间，不连续存放，malloc开辟，空间碎片片）。

（2）查找速度慢,因为查找时,需要循环链表访问,需要从开始节点一个一个节点去查找元素访问。

时间性能：查找O(n) 插入和删除O(1)

线性表的链式表示（真题检测）

1.若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点，则采用（ ）存储方式最节省运算时间。

A.单链表　　B.给出表头指针的单循环链表

C.双链表　　D.带头结点的双循环链表

2.在一个单链表中，若在p所指节点之后插入s所指节点，则执行（）

A.s->next=p;p->next=s　　B.s->next=p->next;p->next=s

C.s->next=p->next;p=s　　D.p->next=s;s->next=p

3.在下列链表中，不能从当前节点出发访问到其余各节点的是　（ ）

A.双向链表　　B.单循环链表

C.单向链表　　D.双向循环链表

答案：D　B　C

线性表的链式表示（真题检测）

4.设带有头结点的单向循环链表的头指针变量为head，则判空条件是 （ ）
A.head==0　　B.head->next==0　、
.head->next==head　　D.head!=0
注释：有的书中写0有的会写NULL 此题为0，大多数书为NULL。灵活运用

5.将长度为n的单链表链接在长度为m的单链表后面，其算法的时间复杂度采用大O形式表示应该是（）
A.O(1)　　B.O(n)　　C. O(m)　　D. O(m+n)

答案：　C　C

线性表的链式表示（真题检测）

6.设线性表中有2n个元素，（）在单链表上实现要比在顺序表上表现效率更高

A.删除所有值为x的元素;

B.在最后一个元素的后面插入如一个新元素;

C.顺序输出前k个元素;

D.交换第i个元素和第2n-i-1个元素的值;

7.在长度为n的有序单链表中插入一个新节点，并仍然保持有序的时间复杂度是（）

A.O(1)　　B.O(n)

C. $O(n^2)$　　D. $O(n\log_2 n)$

答案：A　B

谢 谢 观 看