## 第二章 线性表

一、判断题

1. 顺序表的主要缺点是不利于插入或删除操作。( √ )
2. 线性表采用链式存储时，不同结点的存储空间可以是不连续的。( √ )
3. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。( × )

解析：顺序存储方式的插入、删除运算效率低。

1. 链表是采用链式存储结构的线性表,且进行插入、删除操作时，比在顺序存储结构中效率高。 ( √ )
2. 在线性表的链式存储结构中，逻辑上相邻的两个元素在物理位置上并不一定紧邻。（ √ ）
3. 在线性链表上，删除某结点后，计算机会自动地将后续的各结点向前移动。（ × ）

解析：计算机不会自动地将后续的各结点向前移动。

1. 循环链表的结点结构与单链表的结点结构完全相同，只是结点间的连接方式有不同。（√ ）
2. 顺序表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。（ × ）

解析：顺序表的每个结点可以是一个复杂类型。

1. 线性表链式存储的特点是可以用一组任意的存储单元存储表中的数据元素（ √ ）
2. 线性表采用顺序存储，必须占用一片连续的存储单元。（ √ ）
3. 顺序表结构适宜进行顺序存取，而链表适宜进行随机存取。（ × ）

解析：顺序表经常按位置访问，适宜于顺序存储，进行随机存取，链表适宜于随机存储，顺序存取。

二、选择题

1. 下述哪一条是顺序存储结构的优点？（ A ）

A．存储密度大 B．插入运算方便

C．删除运算方便 D．可方便地用于各种逻辑结构的存储表示

解析：B.C.D是链式存储结构的优点

1. 下面关于线性表的叙述中，错误的是哪一个？（ B ）

A．线性表采用顺序存储，必须占用一片连续的存储单元。

B．线性表采用顺序存储，便于进行插入和删除操作。

C．线性表采用链接存储，不必占用一片连续的存储单元。

D．线性表采用链接存储，便于插入和删除操作。

解析：顺序存储链表不易进行插入和删除操作。

1. 某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（ D ）存储方式最节省运算时间。

A．单链表 B．仅有头指针的单循环链表

C．双链表 D．仅有尾指针的单循环链表

解析：根据题意要求,该线性表的存储应能够很方便地找到线性表的第一个元素和最后一个元素,A和B都能很方便地通过头指针找到钱性表的第一个元素,却要经过所有元素才能找到最后一个元素;选项C双链表若存为双向循环链表,则能很方便地找到线性表的第一个元素和最后一个元素，但存储效率要低些,插入和删除操作也略微复杂; 选项D可通过尾指针直接找到线性表的最后一个元素,通过线性表的最后一个元素的循环指针就能很方便地找到第一个元素。

1. 设一个链表最常用的操作是在末尾插入结点和删除尾结点，则选用( D )最节省时间。

A.单链表 B.单循环链表 C.带尾指针的单循环链表 D.带头结点的双循环链表

解析：带有尾指针的单循环链表删除尾结点时要遍历整个链表，时间复杂度为O（n），只有用带头结点的双向循环链表完成要求的操作最节省时间，时间复杂度是O（1）。

1. 如果某线性表中最常用的操作是取得第i个结点及其前驱，则采用（ D ）存储方式最结省时间。
2. 单链表 B．双向链表 C．单循环链表 D．顺序表

解析：顺序表。采用顺序表存储方式最节省时间，顺序表是在计算机内存中以数组的形式保存的线性表，线性表的顺序存储是指用一组地址连续的存储单元依次存储线性表中的各个元素、使得线性表中在逻辑结构上相邻的数据元素存储在相邻的物理存储单元中，即通过数据元素物理存储的相邻关系来反映数据元素之间逻辑上的相邻关系，采用顺序存储结构的线性表通常称为顺序表。

1. 某线性表中最常用的操作是存取序号为的元素和在最后进行插入和删除运算，则采用( D ) 存储方式时间性能最好。
2. 双向链表 B. 双向循环链表 C. 单向循环链表 D. 顺序表

解析：存储效率高，而且如果是在最后进行插入和删除，顺序表更方便，链表太耗时

1. 静态链表中指针表示的是（ C ）。
2. 内存地址 B．数组下标 C．下一元素地址 D．左、右孩子地址

解析：无

1. 链表不具有的特点是（ B ）。

A．插入、删除不需要移动元素 B．可随机访问任一元素

C．不必事先估计存储空间 D．所需空间与线性长度成正比

解析：链表不具备随机访问，只能循环列表访问

1. 在单链表中，增加头结点的目的是（ C ）。

A.使单链表至少有一个结点 B.标志表中首结点的位置

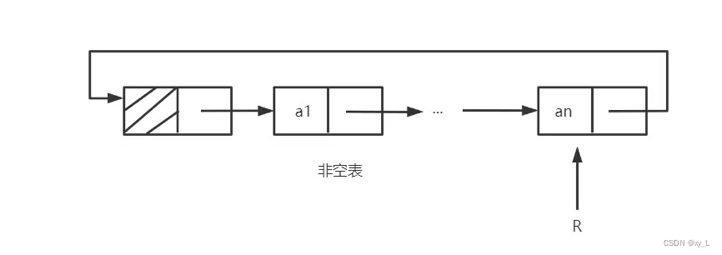
C.方便运算的实现 D.说明该单链表是线性表的链式存储结构

解析：第一，有了头结点后，插入和删除数据的算法统一了，不需要再判断是否第一个元素之前删除或者插入第一个元素。第二，不论链表是否为空，其头指针是指向头结点的空指针，链表的头指针不变，因此空表和非空表的处理也就统一了。

1. 设p为指向单循环链表上某结点的指针，则查找p的直接前驱（ C ）。
2. 找不到 B.的时间复杂度为O(1) C.的时间复杂度为O(n)

解析：头指针表示单循环链表：找a1的时间复杂度是O(1)，找an的时间复杂度是O(n)。

头指针表示单循环链表:

解析：

1. 有一个含头结点的单链表，头指针为head,  则判断其是否为空的条件为( B ).

    A．head==NULL   B.head->next==NULL

1. head->next==head  D. head!=NULL

解析：无

1. 在下列链表中不能从当前结点出发访问到其余各结点的是（ C ）。
2. 双向链表 B.单循环链表 C.单向链表 D.双向循环链表

解析：无

1. 在顺序表中，只要知道（ D ），就可以求出任一结点的存储地址。

Ａ.基地址 B.结点大小 C.向量大小 D.基地址和结点大小

解析：无

1. 对于顺序存储的线性表，访问结点和增加、删除结点的时间复杂度为（ C ）。
2. O(n) O(n) B. O(n) O(1) C. O(1) O(n) D. O(1) O(1)

解析：访问结点只需要找到那个位置即可，但是删除和增加需要把每一个元素都换一个位置

1. 线性表（ a1,a2,…,an）以链接方式存储时，访问第i位置元素的时间复杂度为（ C ）
2. O（i） B．O（1） C．O（n） D．O（i-1）

解析：无

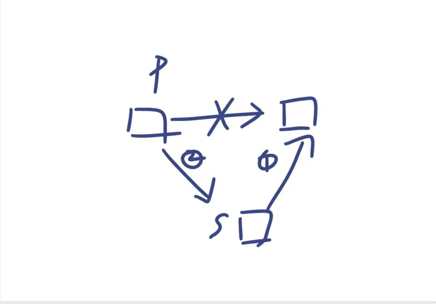
1. 非空的循环单链表head的尾结点p满足（ A ）。
2. p->next==head B．p->next ==NULL C->p== NULL D．p== head

解析：无

1. 在单链表指针为p的结点之后插入指针为s的结点，正确的操作是（ B ）。

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

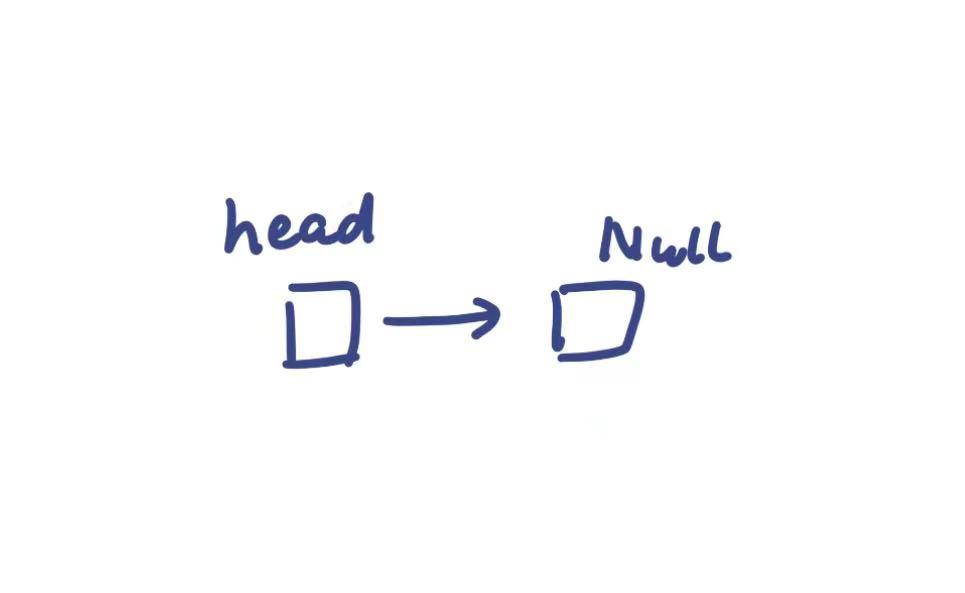
C．p->next=s;p->next=s->next; D．p->next=s->next;p->next=s;

解析：

1. 对于一个头指针为head的带头结点的单链表，判定该表为空表的条件是（ B ）

A．head==NULL B．head->next==NULL

C．head->next==head D．head!=NULL

解析：

1. 完成在双循环链表结点p之后插入s的操作是（ D ）。

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

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

C．s->prior=p; s->next=p->next; p->next=s; p->next->prior:=s ;

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

解析：

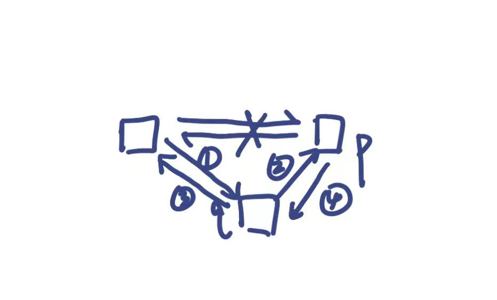
1. 双向链表中有两个指针域，llink和rlink，分别指回前驱及后继，设p指向链表中的一个结点，q指向一待插入结点，现要求在p前插入q，则正确的插入为（ D ）

A.p->llink=q;q->rlink=p; p->llink->rlink=q; q->llink=p->llink;

B.q->llink=p->llink;p->llink->rlink=q;q->rlink=p; p->llink=q->rlink;

C.q->rlink=p; p->rlink=q; p->llink->rlink=q; q->rlink=p;

D.p->llink->rlink=q; q->rlink=p; q->llink=p->llink; p->llink=q;

解析：

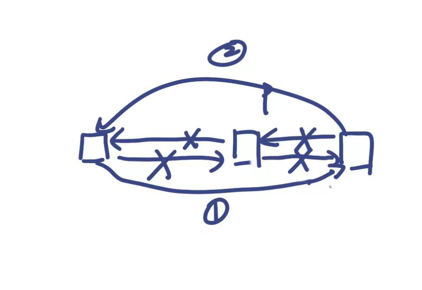
1. 在双向链表存储结构中，删除p所指的结点时须修改指针（ A ）。

A．p->llink ->rlink=p->rlink; p->rlink ->llink=p->llink;

B．p->llink=p->llink ->llink; (p->llink) ->rlink=p;

C．(p->rlink)->llink=p; p->rlink=(p->rlink) ->rlink

D．p->rlink=(p->llink) ->llink; p->llink=(p->rlink) ->rlink;

解析：

1. 设front、rear分别为循环双向链表结点的左指针和右指针，则指针P所指的元素是双循环链表L的尾结点的条件是（ D ）。
2. P= = L B. P->front= =L C. P= =NULL D. P->rear= =L

解析：无

1. 在双向链表中，在已知结点前或后做插入运算的时间复杂度为（ A ）。
2. O(1) B.O(n) C.O(n2) D.O(log2n)

解析：插入或者删除时间复杂度为1

1. 在一个单链表中，已知q所指结点是p所指结点的前驱结点，若在q结点和p结点之间插入s结点，则执行（ C ）。

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

1. q->next=s;s->next=p; D、p->next=s;s->next=q;

解析：无

三、填空题

1. 当线性表的元素总数基本稳定，且很少进行插入和删除操作，要求以最快的速度存取元素时，应采用\_\_\_顺序\_\_\_\_存储结构。

解析：顺序存储很快，但是不利于插入和删除

1. 线性表L=（a1,a2,…,an）用数组表示，假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是\_\_\_\_ （n-1）/2 \_\_\_\_。

解析：删除每个结点的概率都是1/n,删除第一个结点需要移动n-1个结点，删除第二个结点需要移动n-2个结点......删除第n个结点需要移动0个结点。删除一个元素的平均移动元素的个数为：（0+1+2+...+n-1）\*（1/n）。

1. 设单链表的结点结构为(data,next)，next为指针域，已知指针px指向单链表中data为x的结点，指针py指向data为y的新结点 , 若将结点y插入结点x之后，则需要执行以下语句:\_\_\_ py->next=px->next \_\_\_\_； \_ px->next=py \_\_\_\_\_;

解析**：**

1. 在一个长度为n的顺序表中第i个元素（1<=i<=n）之前插入一个元素时，需向后移动\_\_\_\_ \_n-i+1\_\_\_个元素。

解析：无

1. 在双向循环链表中,向p所指的结点之后插入指针f所指的结点，其操作是\_\_\_f->next=p->next\_\_\_\_、\_\_\_\_\_f->prior=p\_\_、\_\_\_\_p->next->prior=f\_\_\_、\_\_\_\_p->next = f\_\_\_\_。

解析：无（可以在选择题那里看图）

1. 在双向链表结构中，若要求在p 指针所指的结点之前插入指针为s 所指的结点，则需执行下列语句：

s->next=p； s->prior= \_\_ p->prior\_\_\_\_\_\_；p->prior:=s；\_\_\_ s->prior->next \_\_\_\_\_=s；

解析：看选择题的图理解

1. 顺序存储结构是通过\_\_\_\_\_\_物理上相邻\_\_表示元素之间的关系的;链式存储结构是通过\_\_\_指针\_\_\_\_\_表示元素之间的关系的。
2. 循环单链表的最大优点是\_\_\_ 从任何一结点出发都可以访问到链表的每一个元素 \_\_\_\_\_。
3. 已知指针p指向单链表L中的某结点，则删除其后继结点的语句是：\_\_\_\_\_u=p->next；p->next=u->next；free（u）\_。

解析：无

1. 在双向链表中要删除已知结点\*P，其时间复杂度为\_ O（1） 。

解析：无

四、简答题

1. 下面是一算法的核心部分，试说明该算法的功能。

pre=L->next; // L是一单链表，结点有数据域 data和指针域 next

if( pre!=NULL )

{

while( pre->next!=NULL)

{ p=pre->next;

if( p->data>=pre->data) pre=p ;

else return(false)

}

return(true); }

解析：该算法的功能是判断链表L是否是非递减有序，若是则返回“true”；否则返回“false”。pre指向当前结点，p指向pre的后继。

1. 设单链表结点指针域为next，试写出删除链表中指针p所指结点的直接后继的C语言语句。

解析：q=p->next; p->next=q->next; free(q);

1. 设单链表中某指针p所指结点（即p结点）的数据域为data，链指针域为next，请写出在p结点之前插入s结点的C语言语句。(单链表的头结点的头指针为head).

解析：pre=head； //单链表的头结点的头指针为head

while(pre->next!=p)

pre=pre->next;

s->next=p;

pre->next=s;

1. 有线性表(a1,a2,…,an),采用单链表存储，头指针为H，每个结点中存放线性表中一个元素，现查找某个元素值等于X的结点。分别写出下面三种情况的查找语句。要求时间尽量少。
2. 线性表中元素无序。（2）线性表中元素按递增有序。 （3）线性表中元素按递减有序。

解析：(1) while(p!=NULL && p->data!=X) p=p->next;

if(p==NULL) return(NULL);∥查找失败

else return(p);∥查找成功

（2）while(p!=NNULL && p->data<X ) p=p->next;

if(p==NULL || p->data>X) return(NULL);∥查找失败

else return(p);

(3) while(p!=NULL && p->data>X) p=p->next;

if(p==NULL || p->data<X) return(NULL); ∥查找失败

else return(p); ∥查找成功

1. 写出下图双链表中对换值为23和15的两个结点相互位置时修改指针的有关语句。结点结构为：(llink,data,rlink)

解析：设 q=p->llink; 则

q->rlink=p->rlink; p->rlink->llink=q;

p->llink=q->llink;q->llink->rlink=p;

p->rlink=q; q->llink=p

1. 设双向循环链表中结点的数据域、前驱和后继指针域分别为data,pre和next,试写出在指针p 所指结点之前插入一s结点的C语言描述语句。

解析：在指针p所指结点前插入结点s的语句如下：

s->pre=p->pre; s->next=p; p->pre->next=s; p->pre=s;

1. 有一线性表，要求重新排列，使所有的正数均在非正数之前，要求用最小交换次数实现，你认为应采用什么方法？

解析：快速排序法，从两头向中间比较，并在必要时进行数据交换，且快速排序是各种排序方法中平均排序速度最快的算法。

1. 有n个非零的数，仅要求将负数排列在正数的前面，但并不要求对其排序，简述处理方法。

解析：快速排序法，从两头向中间比较，并在必要时进行数据交换，且快速排序是各种排序方法中平均排序速度最快的算法。

1. 编写算法，判断带头结点的双循环链表L是否对称。

对称是指：设各元素值a1,a2,...,an, 则有ai=an-i+1

即指：a1= an，a2= an-1  。。。。。。

    结点结构为

|  |  |  |
| --- | --- | --- |
| prior | data | next |

解析：int judge(DLinkList L){

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

while(p!=q)

{if(p->data!=q->data) return 0; if(p->next==q) return 1;

p=p->next;

q=q->prior;

return 1;}