**线性表+栈+队列**

一、选择题

**1.算法的计算量的大小称为算法的()。**

A.效率

B.复杂性

C.现实性

D.难度

**2.算法的时间复杂度取决于( )。**

A.问题的规模

B.待处理数据的初态

C.A和B

**3.计算机算法是指(1)， 它必须具备(2) 这3个特性。**

**(1)**

A.计算方法

B.排序方法

C.解决问题的步骤序列

D.调度方法

**(2)**

A.可执行性、可移植性、可扩充性

B.可执行性、确定性、有穷性

C.确定性、有穷性、稳定性

D.易读性、稳定性、安全性

**4.一个算法应该是( )。**

A.程序

B.问题求解步骤的描述

C.要满足5个基本特性

D. A和C

**5.下面关于算法说法正确的是( )。**

A.算法最终必须由计算机程序实现

B.为解决某问题的算法同为该问题编写的程序含义是相同的

C.算法的可行性是指指令不能有**二义性**

D.以上几个都是错误的

**6.下面说法错误的是( )。**

(1)算法原地工作的含义是指不需要任何额外的辅助空间

(2)在相同的规模n下，复杂度O(n)的算法在时间上总是优于复杂度0(2^n)的算法

(3)所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4)同一个算法，实现语言的级别越高，执行效率就越低

A. (1)

B. (1), (2)

C. (1), (4)

D. (3)

**7.从逻辑上可以把数据结构分为( )两大类。**

A.动态结构、静态结构

B.顺序结构、链式结构(物理上的)

C.线性结构、非线性结构

D.初等结构、构造型结构

**8.下述()与数据的存储结构无关。**

A.栈（**逻辑结构**）

B.双向链表

C.散列表

D.线索树

E.循环队列

**9.在下面的程序段中，对x的赋值语句的时间复杂度为( )。**

**for (i=0; i<n; ++i)**

**for (j=0;j<n; ++j)**

**++x;**

1. O(2n) B. O(n) C. O(n^2) D. 0(log2n)

**10.程序段for(i=n-1;i>=l; --i)**

**for(j=1;j<=i; ++j)**

**if(A[j]>A[j+1]) A[j]与A[j+1]对换;**

**其中n为正整数，则最后一行的语句频度在最坏情况下是( )。**

A. O(n)

B. O(nlog2n)

C. O(n)

D. O(n^2)

N-1,n-2,n-3,n-4,…………，2，1

{（n-1+1）\*(n-1)}/2

**11.以下数据结构中，()是非线性数据结构。**

A.树

B.队

C.栈

**12. 连续存储设计时，存储单元的地址( )。**

A.一定连续

B.一定不连续

C.不一定连续

D.部分连续，部分不连续

**13.以下属于逻辑结构的是( )。**

A.顺序表

B.散列表

C.有序表

D.单链表

# 线性表：

## 1.线性表的存储结构

线性表的存储结构有**顺序存储结构**和**链式存储结构**两种。前者称为**顺序表**，后者称为**链表**。

**顺序表：**

1.顺序表的结构体定义

typedef struct

int data [maxsize] ://存放顺序表元素的数组(默认是int型，可根据题目要求将int换成 其他类型)

int length; //存放顺序表的长度

}Sqlist; //顺序表类型的定义

**int a[maxsize]; //顺序表**

顺序表就是把线性表中的所有元素按照其逻辑顺序，依次存储到从指定的存储位置开始的一块**连续的存储空间**中。这样，线性表中第一个元素的存储位置就是指定的存储位置，第i+1个元素的存储位置紧接在第i个元素的存储位置的后面。

顺序表：随机访问特性。

即顺序表要求**占用连续的存储空间**。存储分配只能预先进行，即**静态分配**，一旦分配好了，在对其操作的过程中始终不变。

顺序表做插入操作的时候要移动多个元素。



**链表：**

1、单链表结点定义

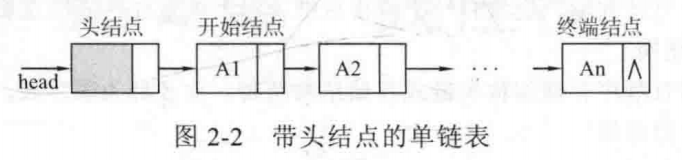
typedef struct LNode

int data; //data中存放结点数据域(默认是int型)

struct LNode "next; //指向后继结点的指针

} LNode; //定义单链表结点类型





2、双链表结点定义

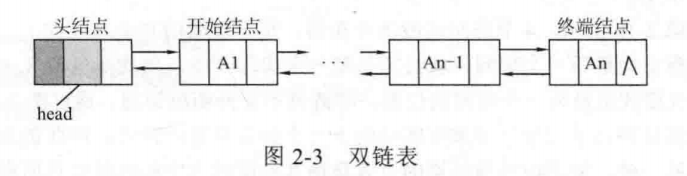
typedef struct DLNode

int data; //data中存放结点数据域(默认是int型)

struct DLNode \*prior; //指向前驱结点的指针

struct DLNode \* next; //指向后继结点的指针

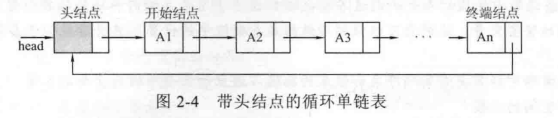
}DLNode;//定义双链表结点类型

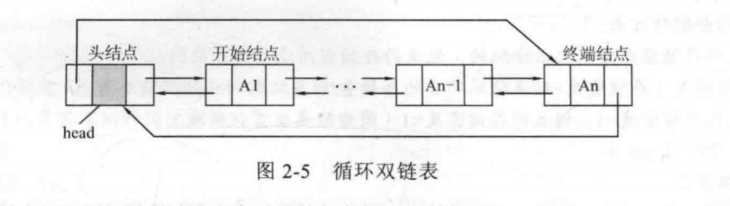


3、循环单链表

**带头结点的循环单链表，当head等于head->next时，链表为空;**

**不带头结点的循环单链表，当head等于NULL时，链表为空。**



1. 循环双链表：
2. 

不带头结点的循环双链表为空，当head等于NULL时。

带头结点的循环双链表中是**没有空指针**的，其空状态下，**head->next**和**head->prior**必然都**等于head**。所以判断其是否为空，只需要检查**heat->next**和**head->prior**两个指针中的**任意一个是否等于head指针**即可。因此，以下四句中的任意一句为真，都可以判断循环双链表为空。

head->next==head;

head->prior==head;

head->next==head& &head->prior==head;

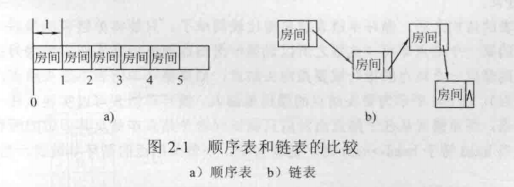
head->next==head I I head->prior==head;

链表不支持随机访问

链表中的每一个结点需要划出一部分空间来存储指向下一个结点位置的指针，因此链表中结点的存储空间利用率较顺序表稍低一些。

链表支持存储空间的动态分配。LNode \*A =(LNode\* ) malloc (sizeof (LNode) ) ;//此句要熟练掌握

在链表中进行插入操作无须移动元素。

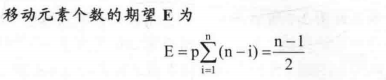


注意：

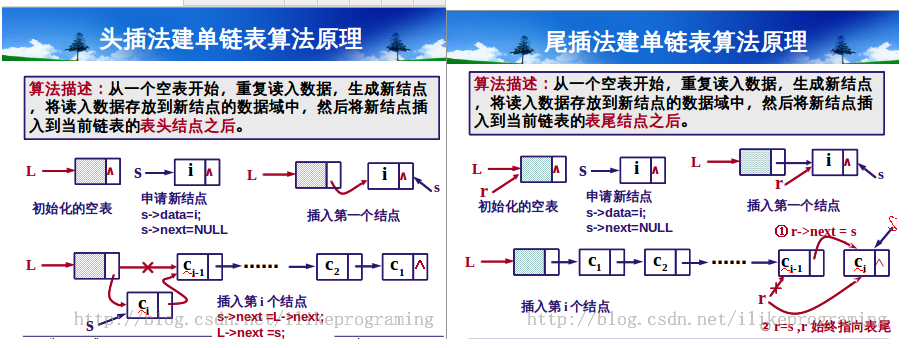
顺序表的存储密度=1，链表的存储密度<1 ( 因为结点中有指针域)。

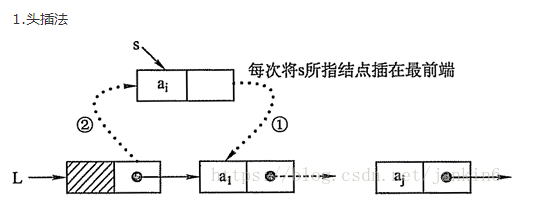
因为插入位置的选择是随机的，所以所有位置被插入的可能性都是相同的，有n个可插入位置，所以任何一个位置被插入元素的概率都为p=1/n。

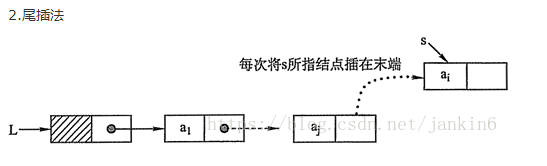
求对应于每个插入位置需要移动的元素个数。假设要把新元素插入在表中第i个元素之后，则需要将第i个元素之后的所有元素往后移动一个位置，因此移动元素个数为n-i.



单链表的插入操作：头插法和尾插法







**具体代码**

**双链表插入结点的算法**

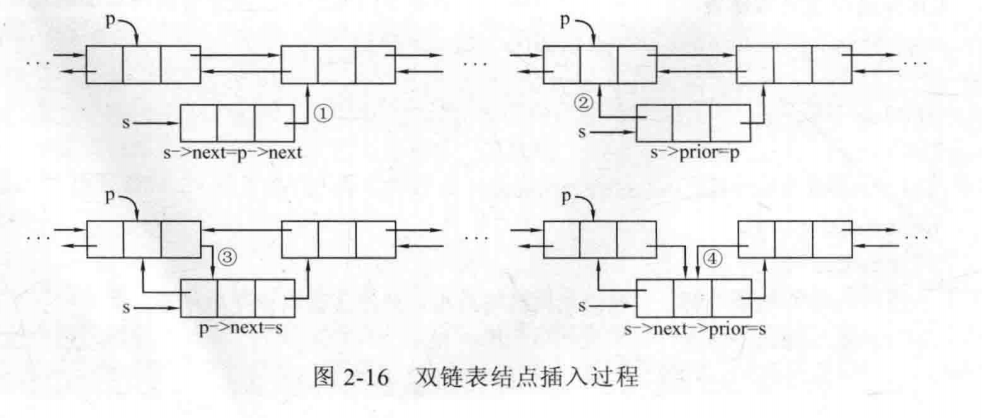
假设在双链表中p所指的结点之后插入一个结点s，其操作语句如下:

s->next=p->next;

s->prior=p;

p->next=s;

s->next->prior=s;//假如p指向最后一个结点，则本行可去掉



**4.删除结点的算法**

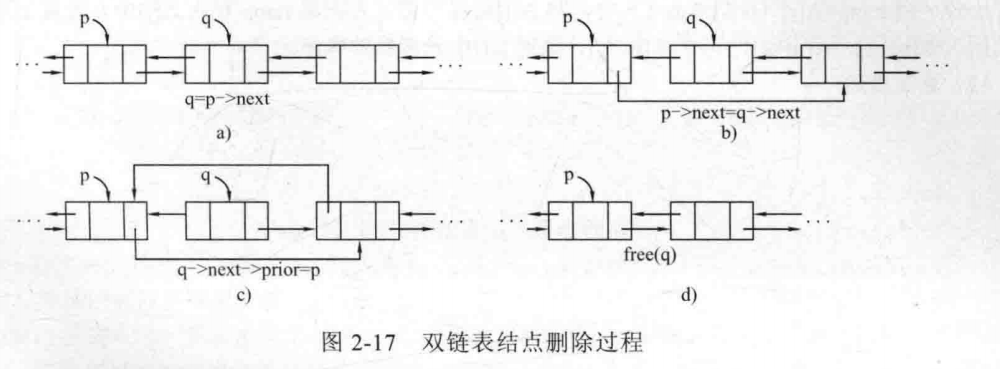
设要删除双链表中p结点的后继结点，其操作语句如下:

q=p->next;

p->next=q->next;

q->next->prior=p;

free(q) ;



**习题**

选择题(题目中的链表如果不特别指出就是带头结点的链表)

**1.下述各项中属于顺序存储结构优点的是()。**

**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

**2.下面关于线性表的叙述中，错误的是()。**

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

**3.线性表是具有n个( )的有限序列。**

A.表元素

B.字符

C.数据元素

D.数据项

**4.若某线性表最常用的操作是存取任一 指定序号的元素和在最后进行插入和删除运算，则利用()存储方式最节省时间。**

A.顺序表

B.双链表

C.双循环链表

D.单循环链表

**5.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( )存储方式最节省运算时间。**

A.单链表

B.不带头结点的单循环链表

C.双链表

D.不带头结点且有尾指针的单循环链表

**6.静态链表中指针指示的是( )。**

A.内存地址

B.数组下标

C.链表中下一元素在数组中的地址

D.左、右孩子地址

**7.链表不具有的特点是( )。**

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

**8.将两个有n个元素的有序表归并成一个有序表，其最少比较次数为( )。**

A. n

B.2n-1

C.2n

D. n-1

**9.单链表L (带头结点)为空的判断条件是( )。**

A. L==NULL

B. L->next==NULL

C. L->next!=NULL

D. L!=NULL

**10.在一个具有n个结点的有序单链表中插入一个新结点仍然保持有序的时间复杂度是()。**

A.0(1)

B. O(n)

C. O(n2)

D. O(nlog2n)

**11.在一个长度为n(n>1)的带头结点的单链表h上，另设有尾指针r (指向尾结点),执行( )操作与链表的长度有关。**

A.删除单链表中的第一个结点

B.删除单链表中的最后一个结点

C.在单链表第一个元素前插入一个新结点

D.在单链表最后-一个元素后插入一一个新结点

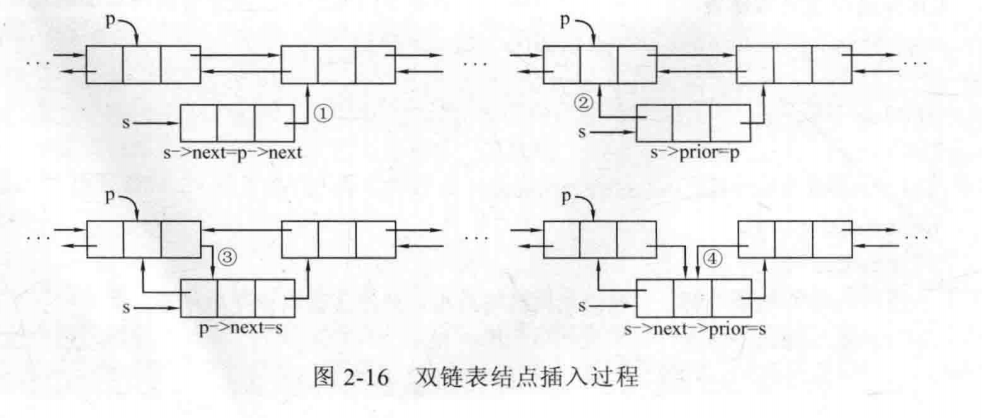
**12.在一个双链表中，在p结点之后插入结点q的操作是( )。**

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

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

C. p->next=q; q->prior=p; q->next=p->next; p->next->prior=q;

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



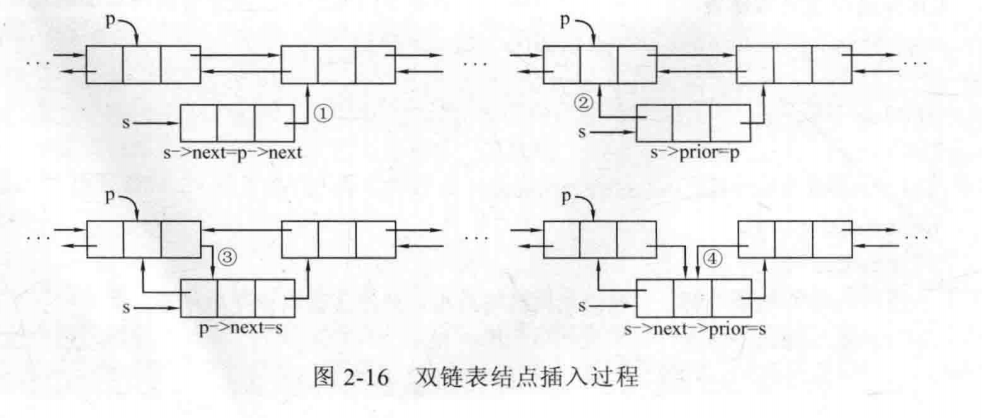
**13.在一个双链表中，在p结点之前插入q结点的操作是()。**

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

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

C. q->next=p; p->next =q; q->prior->next=q; q->next=P;

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



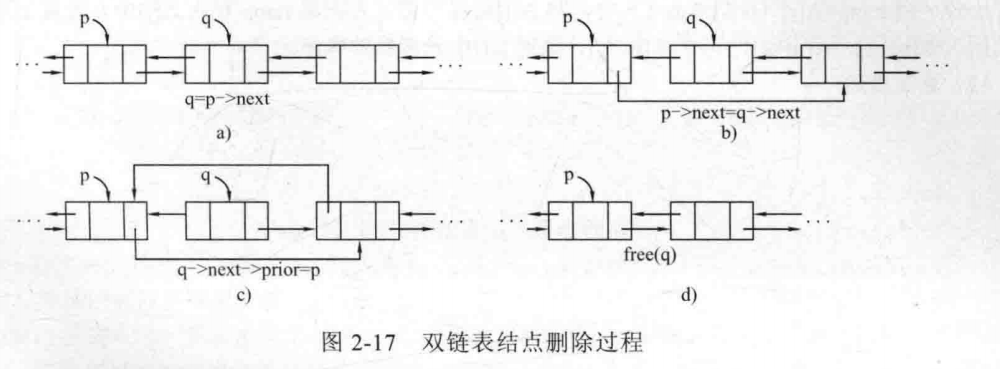
**14.在一个双链表中，删除p结点的操作是( ) (结点空间释 放语句省略)。**

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

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

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

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



**15.非空的单循环链表L (带头结点)的终端结点(由p所指向)满足( )。**

A. p->next==NULL

B. p==L

C. p->next==L

D. p->next==L&&p!=L

**16.带头结点的双循环链表L为空的条件是( )。**

A. L==NULL

B. L->next->prior==NULL

C. L->prior= =NULL

D. L->prior==L&&L->next==L

**17.线性表是()。**

A.一个有限序列，可以为空

B.一个有限序列，不可以为空

C.一个无限序列，可以为空

D.一个无限序列，不可以为空

**18. 线性表采用链表存储时，其地址()**

A.必须是连续的

B.--定是不连续的

C.部分地址必须是连续的

D.连续与否均可以

**19.线性表的静态链表存储结构与顺序存储结构相比，其优点是()。**

A.所有的操作算法实现简单

B.便于随机存取

C.便于插入和删除

D.便于利用零散的存储空间

**20.设线性表有n个元素，以下操作中，( ) 在顺序表上实现比在链表上实现效率更高。**

A.输出第i (1≤i≤n)个元素值

B.交换第1个元素与第2个元素的值

C.顺序输出这n个元素的值

D.输出与给定值x相等的元素在线性表中的序号

**21.对于一个线性表，既要求能够快速地进行插入和删除，又要求存储结构能够反映数据元素之间**

**的逻辑关系，则应采用( ) 存储结构。**

A.顺序

B.链式

C.散列(Hash表)

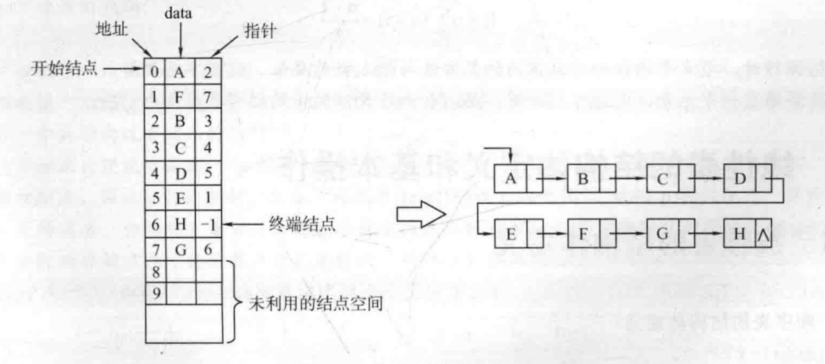
**22.需要分配较大的连续空间，插入和删除不需要移动元素的线性表，其存储结构为()。**

A.单链表

B.静态链表

C.顺序表a[i][j][k]

D.双链表



**23.如果最常用的操作是取第i个元素的前驱结点，则采用()存储方式最节省时间。**

A.单链表

B.双链表

C.单循环链表

D.顺序表

**24.与单链表相比，双链表的优点之一是( )。**

A.插入、删除操作更简单

B.可以进行随机访问

C.可以省略表头指针或表尾指针

D.访问前后相邻结点更灵活

**25.在顺序表中插入一个元素的时间复杂度为(一)。**

A.0(1)

B. O(log2n)

C. O(n)

D. O(n2)

**26.在顺序表中删除--一个元素的时间复杂度为()。**

A.0(1)

B. O(log2n)

C. O(n)

D. O(n2)

**27.对于一个具有n个元素的线性表，建立其单链表的时间复杂度为( )。**

A.0(1)

B. O(log2n)

C. O(n)

D. O(n^2)

**28.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() )**

**存储结构最节省运算时间。**

A.单链表

B.给出表头指针的循环单链表

C.双链表

D.带头结点的循环双链表

**29.线性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第一个结点，则采用( )**

**存储方式最节省时间。**

A.单链表

B.仅有头结点的单循环链表

C.双链表

D.仅有尾结点指针的单循环链表

**30.设有两个长度为n的单链表(带头结点),结点类型相同，若以hI为头结点指针的链表是非循**

**环的，以h2为头结点指针的链表是循环的，则( )。**

A.对于两个链表来说，删除开始结点的操作，其时间复杂度分别为0(1)和0(n)

B.对于两个链表来说，删除终端结点的操作，其时间复杂度都是O(n)

C.循环链表要比非循环链表占用更多的内存空间

D. h1和h2是不同类型的变量

**栈的基本概念**

1.栈的定义

栈是一种只能在一端进行插入或删除操作的**线性表**。其中允许进行插入或删除操作的-端称为**栈顶**

**(Top)。**栈顶由一个称为栈顶指针的位置指示器(其实就是一个变量，对于顺序栈，就是记录栈项元素所

在数组位置标号的一个整型变量:对于链式栈，就是记录栈顶元素所在结点地址的指针)来指示，它是

动态变化的。表的另一端称为**栈底**，栈底是固定不变的。栈的插入和删除操作一般称为**入栈和出栈**。

栈可以依照存储结构分为两种:顺序栈和链式栈。

1.顺序栈定义

typedef struct

int data [maxSize] ;//存放栈中元素，maxSize是已定义的常量

int top;//栈顶指针

} SqStack;//顺序栈类型定义

2.链栈结点定义

typedef struct LNode

{

int data;//数据域

struct LNode \*next; //指针域

} LNode;//链栈结点定义

链栈就是采用链表来存储栈。这里用带头结点的单链表来作为存储体，

图3-1 顺序栈示意图

3.1.2

队列的基本概念

1.队列的定义

队列简称队，它也是一-种操作受限的线性表，其限制为仅允许在表的一端进行插入， 在表的另一端

进行删除。可进行插入的一端称为队尾(Rear), 可进行删除的一端称为队头(Front)。 向队列中插入新

的元素称为进队，新元素进队后就成为新的队尾元素;从队列中删除元素称为出队，元素出队后，其后

继元素就成为新的队头元素。

2.队列的特点

队列的特点概括起来就是先进先出(FIFO)。 打个比方，队列就好像开进隧道的一列火车，各节车

厢就是队中的元素，最先开进隧道的车厢总是最先驶出隧道。

3.队列的存储结构

可用顺序表和链表来存储队列，队列按存储结构可分为顺序队和链队两种。