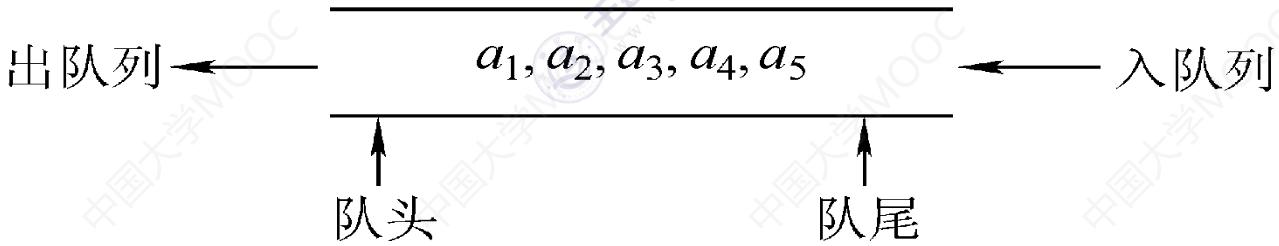
队列
（Queue）

队列（Queue）简称队，也是一种操作受限的线性表，只允许在表的一端进行插入，而在表的另一端进行删除。向队列中插入元素称为入队或进队；删除元素称为出队或离队。

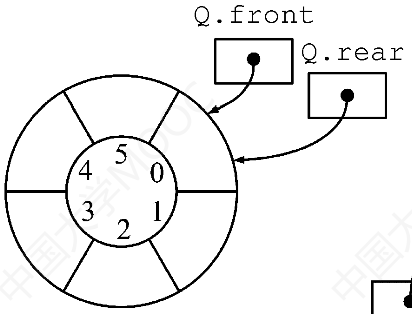
队头（Front）。允许删除的一端，又称队首。
队尾（Rear）。允许插入的一端。



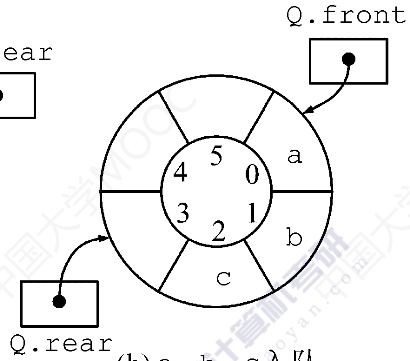
特性是先进先出（**First In First Out，FIFO**）

循环队列

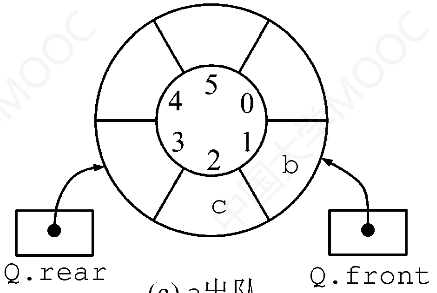
```
#define MaxSize 5
typedef int ElemType;
typedef struct{
    ElemType data[MaxSize];//数组,存储MaxSize-1个
    元素
    int front,rear;//队列头 队列尾
}SqQueue;

SqQueue Q;
```



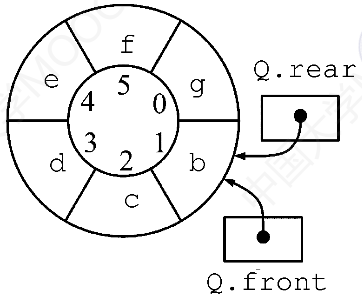
(a) 初始空队



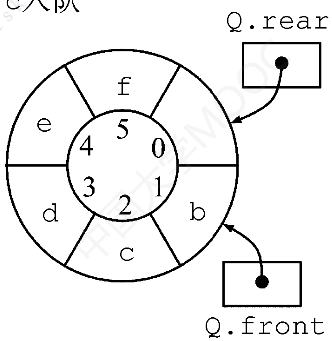
(b) a、b、c入队



(c) a出队



(d1) d、e、f、g入队
（无法判断队满还是队空）



(d2) d、e、f入队
（牺牲一个存储单元）

元素入队

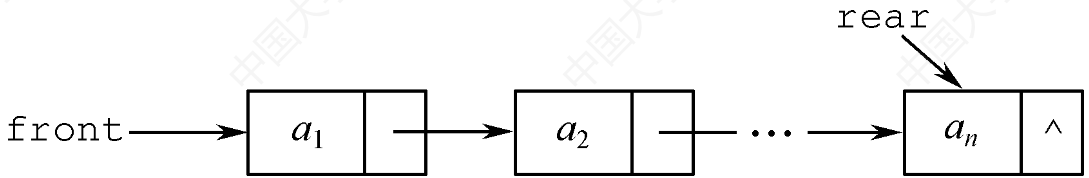
```
bool EnQueue(SqQueue &Q,ElemType x)
{
    if((Q.rear+1)%MaxSize==Q.front) //判断是否队满
        return false;
    Q.data[Q.rear]=x;//放入元素
    Q.rear=(Q.rear+1)%MaxSize;//改变队尾标记
    return true;
}
```

元素出对

```
bool DeQueue(SqQueue &Q,ElemType &x)
{
    if(Q.rear==Q.front)
    return false;
    x=Q.data[Q.front];//先进先出
    Q.front=(Q.front+1)%MaxSize;
    return true;
}
```

队列的链式存储

队列的链式表示称为链队列，它实际上是一个同时带有队头指针和队尾指针的**单链表**。头指针指向队头结点，尾指针指向队尾结点，即单链表的最后一个结点。



存储结构

```
typedef int ElemType;
typedef struct LinkNode{
        ElemType data;
        struct LinkNode *next;
}LinkNode;
typedef struct{
     LinkNode *front,*rear;//链表头 链表尾
}LinkQueue;//先进先出

LinkQueue Q;
```

相对于原有编写的链表增加了尾指针

分为初始化队列---依然带有头结点
入队
出队
判断队列是否为空

斐波那契序列

与之前上台阶的区别是f(2)=1