

## 2.3.1队列的顺序存储及实现

---

什么是队列？**队列（Queue）**：具有一定操作约束的线性表

**插入和删除操作**：只能在一端插入，而在另一端删除。

- 数据插入：入队列（**AddQ**）
- 数据删除：出队列（**DeleteQ**）
- 先来先服务
- 先进先出：FIFO

---

队列的抽象数据类型描述

**类型名称**：队列（Queue）

**数据对象集**：一个有0个或多个元素的有穷线性表

**操作集**：长度为MaxSize的队列，队列元素

1. Queue CreateQueue(int MaxSize):生成长度为MaxSize的空队列
2. int IsFullQ(Queue Q,int MaxSize):判断队列Q是否已满
3. void AddQ(Queue Q,ElementType item):将数据元素item插入队列中
4. int IsEmptyQ(Queue Q):判断队列Q是否为空
5. ElementType DeleteQ(Queue Q):将队头数据元素从队列中删除并返回

---

**队列的顺序存储实现**

队列的顺序存储结构通常由一个**一维数组**和一个记录队列头元素位置的变量front以及一个记录队列尾元素位置的变量rear组成。

假设最开始front和rear都是-1，每增加一个队列元素，rear+1，每减少一个队列元素，front+1

**顺环队列**

1. 这种方案：堆栈空和满的判别条件是什么？
2. 为什么会出现空、满无法区分？根本原因？

如果说数组大小为n的话，front和rear之间的差距有n种情况，而装载数据的情况却有 种（不装和装满两者必须舍弃其中一个）

**解决方案**

1. 使用额外标记：Size或者tag域
  1. Size用于记录当前元素个数
  2. 加入一个元素tag设为1，删除一个元素tag设为0，最后如果front和rear搞不清楚到底是空还是满就去看tag，tag表示了最后一次操作是加入还是删除
2. 仅使用 个数组空间

队列的顺序存储实现

```

#define MaxSize 1000
struct QNode
{
    ElementType data[MaxSize];
    int rear;
    int front;
};
typedef struct QNode *Queue;

/* 1.入队列 */
void AddQ(Queue PtrQ,ElementType item)
{
    if((PtrQ->rear+1)%MaxSize==PtrQ->front)
    {
        printf("队列满");
        return ;
    }
    PtrQ->rear=(PtrQ->rear+1)%MaxSize;
    PtrQ->data[PtrQ->rear]=item;
}

/* 2.出队列 */
ElementType DeleteQ(Queue PtrQ)
{
    if(PtrQ->front==PtrQ->rear)
    {
        printf("队列空");
        return ERROR;
    }
    else
    {
        PtrQ->front=(PtrQ->front+1)%MaxSize;
        return PtrQ->data[PtrQ->front];
    }
}

```

### 2.3.2队列的链式存储实现

---

队列的链式存储结构也可以用一个单链表实现。插入和删除操作分别在链表的两头进行；队列指针**front**和**rear**应该分别指向链表的哪一头？

- front要做删除操作，rear要做插入操作

队列的链式存储实现

```

struct Node
{
    ElementType Data;
    struct Node *Next;
};
struct QNode /* 链队列结构 */
{
    struct Node *rear; /* 指向队尾结点 */
    struct Node *front; /* 指向队头结点 */
};
typedef struct QNode *Queue;
Queue PtrQ;

/* 不带头结点的链式队列出队操作的一个示例 */

ElementType DeleteQ(Queue PtrQ)
{
    struct Node *FrontCell;
    ElementType FrontElem;
    if(PtrQ->front==NULL)
    {
        printf("队列空");
        return ERROR;
    }
    FrontCell=PtrQ->front;
    if(PtrQ->front==PtrQ->rear) /* 若队列只有一个元素 */
    {
        PtrQ->front=PtrQ->rear=NULL; /* 删除后队列置为空 */
    }
    else
    {
        PtrQ->front=PtrQ->front->Next;
    }
    FrontElem=FrontCell->Data;
    free(FrontCell); /* 释放被删除结点空间 */
    return FrontElem;
}

```

如何用两个堆栈实现一个队列？如果堆栈容量分别为\_\_\_\_\_，队列容量为？

答：定义两个堆栈，一个堆栈用于入队，当要入队时，按照顺序压入堆栈1。当要出队时，检查堆栈2是否有元素，若有元素则直接弹出堆栈2的栈顶元素。若无元素，则将堆栈1中所有元素弹出并逐个压入堆栈2中，并弹出栈顶元素。