# 概述

所谓的队列是一种先入先出（FIFO，First In First Out）的数学模型。

# 2、分类

## 普通队列

普通队列对于队列的长度有一个限制，如果内存不能继续分配时，则队尾不能继续向后增加。

普通队列有两种处理方式：

1. 队头出去以后后面的元素依次向前移动一个位置，这样元素移动多处理

效率低；

1. 队列元素不移动，只移动队头指针，这样比较浪费空间资源。

## 环形队列

为了弥补普通队列的缺点，引入了环形队列。

环形队列的排队是有顺时针和逆时针之说的。

在元素出队后，队头指针是不断向后移动的，队尾也是可以向后移动的，这样处理速度快，而且也可以充分利用空间。

# 3、普通队列实现

## 顺序存储

### 定义

typedef struct SeqQueue{

int data[MAXSIZE];

int front,rear;

}SeqQueue;

### 获取元素

int SeqQueueGetElement(SeqQueue \*q,int \*e){

\*e=q->data[q->front];

q->front--;

}

### 插入元素

int SeqQueueEnQueue(SeqQueue \*q,int \*e){

q->data[q->rear]=\*e;

q->read=(q->rear+1)%MAXSIZE;

return 0;

}

### 删除元素

int SeqQueueDeQueue(SeqQueue \*q,int \*e){

\*e=q->data[q->front];

q->front=(q->front+1)%MAXSIZE;

return 0;

}

## 链式存储

# 4、环形队列实现