

小马加编信息学教案(十九)

队列

- [一. 课程内容](#)
- [二. 知识讲解](#)
 - [1. 队列概念](#)
 - [2. 队列基本操作实现](#)
 - * [2.1 判断队列是否为空 \(empty\)](#)
 - * [2.2 清空 \(clear\)](#)
 - * [2.3 队尾插入 \(push\)](#)
 - * [2.4 队头查看 \(front\)](#)
 - * [2.5 队头删除 \(pop\)](#)
 - * [2.6 循环队列实现](#)
- [三. 经典例题](#)
- [四. 提高巩固](#)

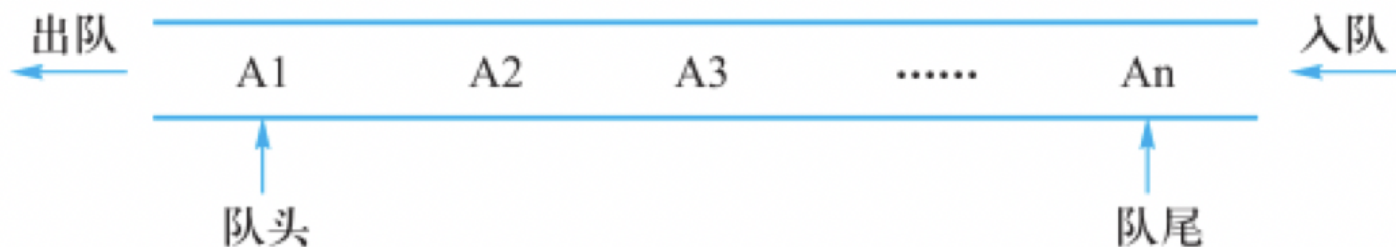
一. 课程内容

1. 队列概念
2. 队列基本操作实现

二. 知识讲解

1. 队列概念

队列是一种数据结构，可以理解为组织变量以存储数据的一种方式
可以把队列理解为一个规定了头、尾的变量序列
但规定只能对序列的两端的变量进行存储或删除的操作，比如



我们规定只允许

查询当前队头元素（即查询 A_1 ）

删除队头元素（即删除 A_1 , 然后 A_2 成为队头）

在当前队尾后加入元素（即在 A_n 后加入 A_{n+1} , 然后 A_{n+1} 成为队尾）

这样的数据结构看起来很鸡肋是不是。。还没有数组好用

其实有些算法的实现只需要队列的功能，所以还是有必要学习的

2. 队列基本操作实现

我们采用一维数组A来实现队列

令变量front指向队头前一个位置（即队头为 $A[\text{front}+1]$ ）

变量rear指向队尾

初始时令 $\text{front}=\text{rear}=0$

暂时不考虑数组下标溢出的问题

假设元素为int（其它类型同理）

2.1 判断队列是否为空（empty）

```
bool empty()
{
    if(front==rear) return 1;
    return 0;
}
```

为空返回1，非空返回0

2.2 清空（clear）

```
void clear()
{
    front=rear;
}
```

队列元素为 $A[\text{front}+1] \sim A[\text{rear}]$ ，当 $\text{front}=\text{rear}$ 时队列为空``

2.3 队尾插入（push）

```
void push(int x)
{
    A[++rear]=x;
}
```

>在队尾后插入元素x，并使元素x成为队尾

2.4 队头查看 (front)

```
int front()
{
    return A[front+1];
}
```

返回值为队头元素

在查看前先判断队列是否为空，队列为空时查看队头无意义

2.5 队头删除 (pop)

```
void pop()
{
    front++;
}
```

在查看前先判断队列是否为空，队列为空时删除队头无意义，不执行

2.6 循环队列实现

在之前的实现中，我们总假设数组大小是足够的，不会越界

然而在实际情况中，我们可能要把数组开得很大才能保证

但是如果已知任何时刻队列中元素不会超过n个，我们只需要在每次操作之后执行

```
if(rear==n) rear=0;
if(front==n) front=0;
```

即可

这样数组大小只需要n

因为这样相当于把队列变为一个循环的圈，A[n]后面的位置是A[1]

又由于队列中最后有n个元素，所以不会造成覆盖，不会丢失元素

队列特点：**先进先出**

三. 经典例题

1. 在纸上模拟一个存储int的队列。
给出每个有返回值操作的返回值。
并画出每个操作后的队列。

```
push 1
push 7
push 5
```

```
front
pop
push 4
front
empty
clear
empty
```

2. 使用c++实现一个储存int的队列。
给出每个有返回值操作的返回值。

输入格式：同上题

输出格式：每行一个整数，表示对应返回值

样例输入	样例输出
empty push 5 push 7 front pop front empty	 1 5 7 0

- 3.使用循环队列实现上一题

四. 提高巩固

- 1.周末舞会

在周末舞会上，男士们和女士们进入舞厅时，各自排成一队。跳舞开始时，依次从男队和女队的队头上各出一人配成舞伴。规定每个舞曲只能有一对跳舞者。跳完舞后男女各自回到队尾。现要求写一个程序，模拟上述舞伴配对问题。

输入格式: 第 1 行两个正整数，表示男士人数 m 和女士人数 n ， $1 \leq m, n \leq 1000$ ；
第 2 行一个正整数，表示舞曲的数目 k ， $k \leq 1000$ 。

输出格式: 共 k 行，每行两个数，之间用一个空格隔开，表示配对舞伴的序号，男士在前，女士在后。

样例输入	样例输出
------	------

样例输入	样例输出
2 4 6	1 1 2 2 1 3 2 4 1 1 2 2

2.取牌游戏

小明正在使用一堆共 K 张纸牌与 $N-1$ 个朋友玩取牌游戏。其中， $N \leq K \leq 100000$ ， $2 \leq N \leq 100$ ， K 是 N 的倍数。纸牌中包含 $M=K/N$ 张“good”牌和 $K-M$ 张“bad”牌。小明负责发牌，他当然想自己获得所有“good”牌。

他的朋友怀疑他会欺骗，所以他们给出以下一些限制，以防小明耍诈：

- 1) 游戏开始时，将最上面的牌发给小明右手边的人。
- 2) 每发完一张牌，他必须将接下来的 P 张牌 ($1 \leq P \leq 10$) 一张一张地依次移到最后，放在牌堆的底部。
- 3) 以逆时针方向，连续给每位玩家发牌。

小明迫切想赢，请你帮助他算出所有“good”牌放置的位置，以便他得到所有“good”牌。牌从上往下依次标注为1, 2, 3, ...

输入格式:第1行，3个用一个空格间隔的正整数 N 、 K 和 P

输出格式:1行 M 个数，从顶部按升序依次输出“good”牌的位置

样例输入	样例输出
3 9 2	3 7 8

3.海港(noip2016普及组)

小K是一个海港的海关工作人员，每天都有许多船只到达海港，船上通常有很多来自不同国家的乘客。

小K对这些到达海港的船只非常感兴趣，他按照时间记录下了到达海港的每一艘船只情况；

对于第 i 艘到达的船，他记录了这艘船到达的时间 t_i (单位：秒)，船上的乘客数 k_i ，以及每名乘客的国籍 $x_{i,1}, x_{i,2}, \dots, x_{i,k}$

小K统计了 n 艘船的信息，希望你帮忙计算出以每一艘船到达时间为止的24小时(24小时=86400秒)内所有乘船到达的乘客来自多少个不同的国家。

形式化地讲，你需要计算 n 条信息。对于输出的第 i 条信息，你需要统计满足 $t_i - 86400 < t_p < t_i$ 的船只 p ，在所有的 $x_{p,j}$ 中，总共有多少个不同的数。

输入格式：

第一行输入一个正整数 n ，表示有 n 艘船的信息。

接下来n行，每行描述一艘船的信息：前两个整数 t_i 和 k_i 分别表示这艘船到达海港的时间和船上的乘客数量，接下来 k_i 个整数 $x_{i,j}$ 表示船上乘客的国籍。

保证输入的 t_i 是递增的，单位是秒；表示从小K第一次上班开始计时，这艘船在第 t_i 秒到达海港。

保证 $1 \leq n \leq 10^5$, $\sum k_i \leq 3 * 10^5$,

$1 \leq x(i, j) \leq 10^5, 1 \leq t(i - 1) \leq t_i \leq 10^9$

其中 $\sum k_i$ 表示所有的 k_i 的和。

输出格式：

输出n行，第i行输出一个整数表示第i艘船到达后的统计信息。

样例输入	样例输出
3 1 4 4 1 2 2 2 2 2 3 10 1 3	3 4 4