

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



队列的顺序存储结构

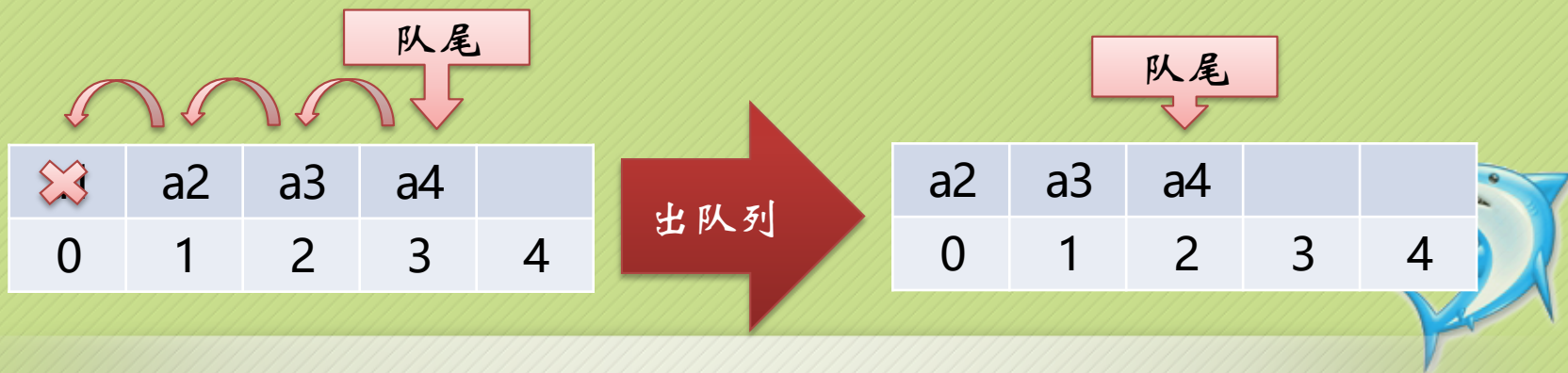
- 为什么小甲鱼上节课说队列的实现上我们更愿意用链式存储结构来存储?
- 我们先按照应有的思路来考虑下如何构造队列的顺序存储结构, 然后发掘都遇到了什么麻烦。
- 我们假设一个队列有 n 个元素, 则顺序存储的队列需建立一个大于 n 的存储单元, 并把队列的所有元素存储在数组的前 n 个单元, 数组下标为0的一端则是队头。
- No Pic You Say a J8



队列的顺序存储结构



入队列操作其实就是在队尾追加一个元素，不需要任何移动，时间复杂度为 $O(1)$ 。出队列则不同，因为我们已经架设下标为0的位置是队列的队头，因此每次出队列操作所有元素都要向前移动。



队列的顺序存储结构

- 在现实中也是如此，一群人在排队买火车票，前边的人买好了离开，后面的人就要全部向前一步补上空位。
- 可是我们研究数据结构和算法的一个根本目的就是要想方设法提高我们的程序的效率，按刚才的方式，出队列的时间复杂度是 $O(n)$ ，效率大打折扣！
- 如果我们不去限制队头一定要在下标为0的位置，那么出队列的操作就不需要移动全体元素。



队列的顺序存储结构



- 但是这样也会出现一些问题，例如按下边的情形继续入队列，就会出现数组越界的错误。



- 可事实上我们有0和1两个下标还空着，这叫假溢出。

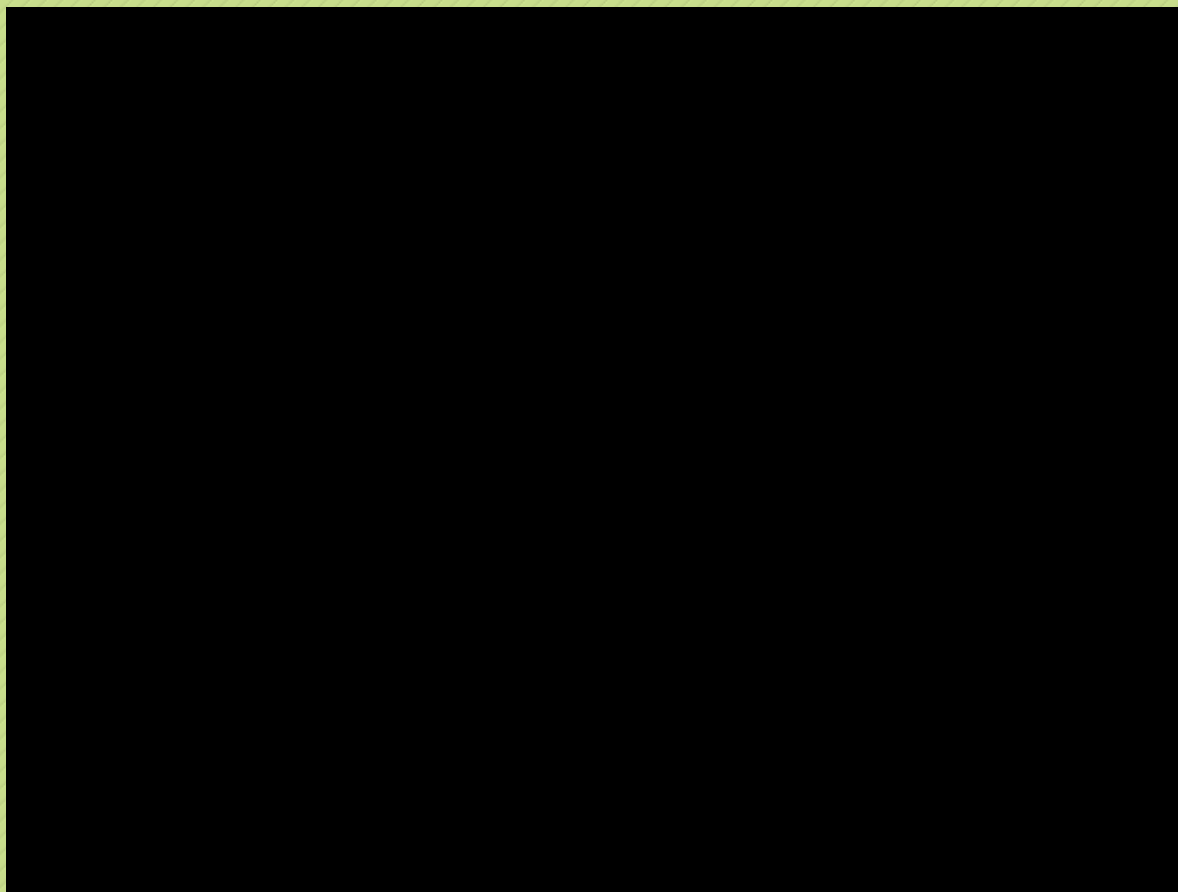


循环队列定义

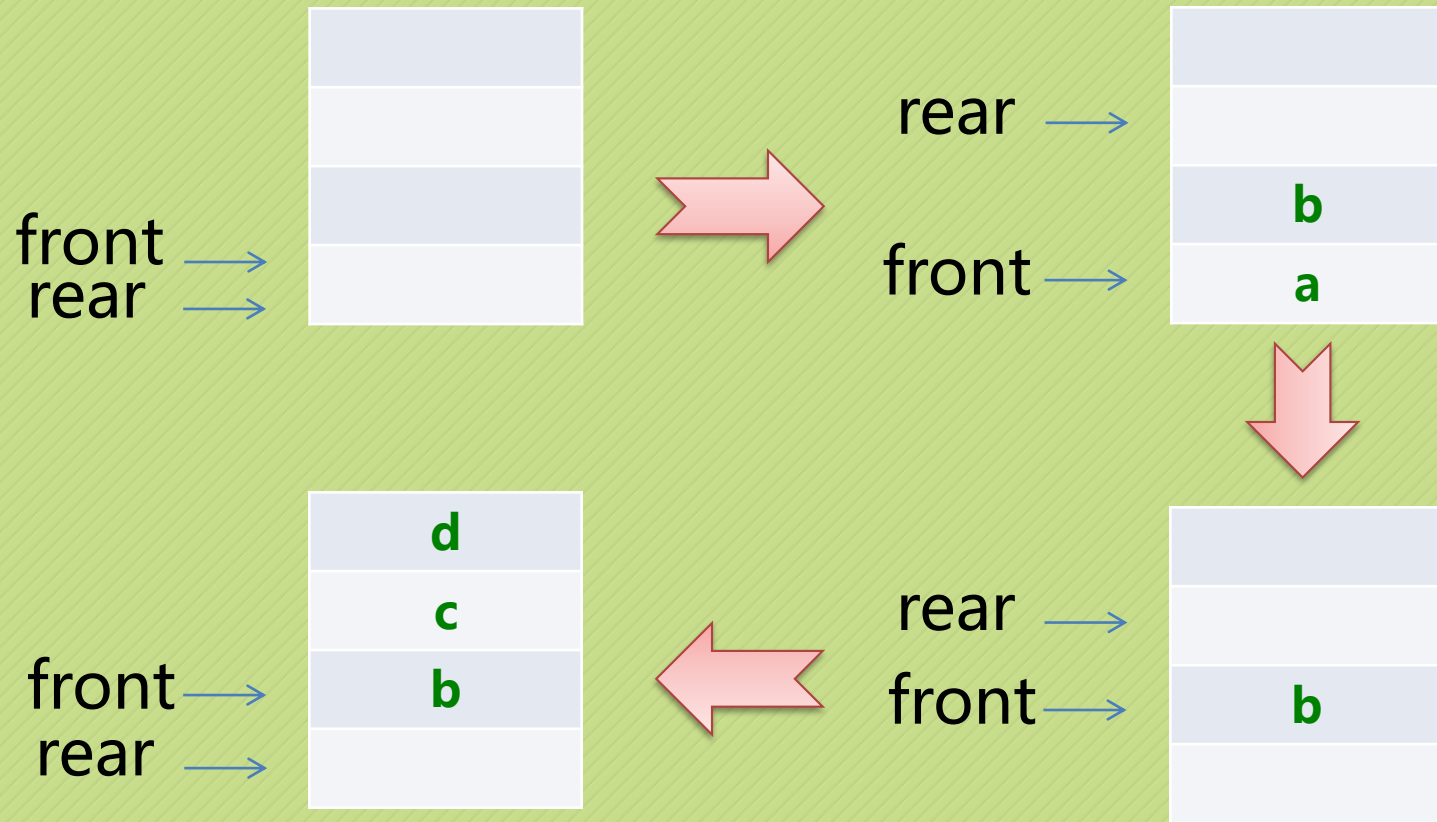
- 我们再想想，要解决假溢出的办法就是如果后面满了，就再从头开始，也就是头尾相接的循环。
- 循环队列它的容量是固定的，并且它的队头和队尾指针都可以随着元素入出队列而发生改变，这样循环队列逻辑上就好像是一个环形存储空间。
- 但要注意的是，在实际的内存当中，不可能有真正的环形存储区，我们只是用顺序表模拟出来的逻辑上的循环。
- 我们通过一段动画片来加深印象吧！



循环队列定义



循环队列定义



循环队列定义

- 于是我们发觉了，似乎循环队列的实现只需要灵活改变front和rear指针即可。
- 也就是让front或rear指针不断加1，即时超出了地址范围，也会自动从头开始。我们可以采取取模运算处理：
 - $(\text{rear}+1) \% \text{QueueSize}$
 - $(\text{front}+1) \% \text{QueueSize}$
- 取模就是取余数的意思，他取到的值永远不会大于除数，大家结合实例拿张纸算一算就知道啦~



代码清单

- 定义一个循环队列

```
#define MAXSIZE 100
```

```
typedef struct
```

```
{
```

```
    ElemType *base; // 用于存放内存分配基地址
```

```
    // 这里你也可以用数组存放
```

```
    int front;
```

```
    int rear;
```

```
}
```



代码清单

- 初始化一个循环队列

```
initQueue(cycleQueue *q)
{
    q->base = (ElemType *) malloc (MAXSIZE *
    sizeof(ElemType));
    if( !q->base )
        exit(0);
    q->front = q->rear = 0;
}
```



代码清单

- 入队列操作

```
InsertQueue(cycleQueue *q, ElemType e)
{
    if( (q->rear+1)%MAXSIZE == q->front )
        return; // 队列已满
    q->base[q->rear] = e;
    q->rear = (q->rear+1) %MAXSIZE;
}
```



代码清单

- 出队列操作
 - DeleteQueue(cycleQueue *q, ElemType *e)
 - {
 - if(q->front ==q->rear)
 - return; // 队列为空
 - *e =q->base[q->front];
 - q->front = (q->front+1) %MAXSIZE;
 - }



人生如栈

- 人生，就像是栈的演变。
- 在父亲忙碌的入栈、出栈操作中，你，诞生了！
- 人生，仿佛是栈的重现。
- 每天你奔波于事业与家庭之间，做着似乎总是重复的事情，为的只是一餐温饱。
- 你总说，在哪里跌倒就在哪里爬起来，但是，你发现似乎总是在同一个地方跌倒无数次！
- 有一那么一次，你弹栈找不到返回地址，你忧郁了，迷茫了。。。。。



人生如栈

- 很幸运，你看到了小甲鱼的视频^_^
- 他会告诉你，在一个个栈的外边，其实隐藏着一个队列在调用你的每一个栈，只是你还年轻，没办法看得太清楚。
- 但是，你抬头仰望星空，在孤独中多思考，在彷徨中多回顾，就有希望，不断进取，就能成功！
- 人生，又是一个大队列的实现，春夏秋冬年复一年，改变的是时间，不变的是你对未来执着的信念！



人生如栈



- 地球近似于圆形，但并不完美。
- 选错了方向，你可能要多走一些路，
- 但，只要你肯坚持到底，你都可以到达终点！

