

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



神马是队列

- 这就是队列



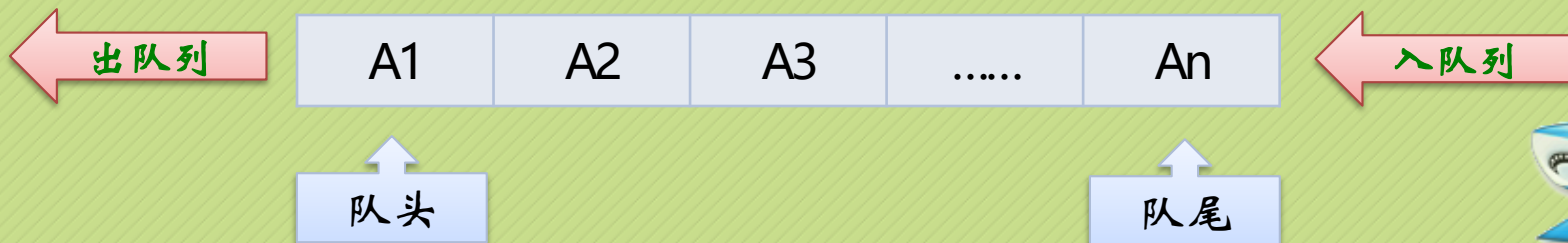
神马是队列

- 这不是队列



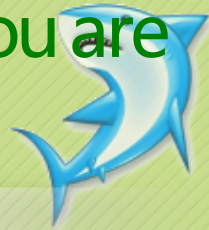
队列的定义

- 队列 (queue) 是只允许在一端进行插入操作, 而在另一端进行删除操作的线性表。
- 与栈相反, 队列是一种先进先出 (First In First Out, FIFO) 的线性表。
- 与栈相同的是, 队列也是一种重要的线性结构, 实现一个队列同样需要顺序表或链表作为基础。



队列的定义

- 此前我们用浏览器的历史记录作为栈的例子让大家了解她的应用广泛，队列在现实中也是应用十分广泛。
- 我们的输入缓冲区接受键盘的输入就是按队列的形式输入和输出的，不然的话就很可能闹出问题。
- 例如有一天你突然心血来潮，用企鹅发了一句
“You are my god” 给你女朋友，表示她就是你的全部，但是输入缓冲区在输入god这个单词的时候不用队列这个结构而用栈的结构，就变成了“You are my dog”发出去了。。。。。。



队列的链式存储结构

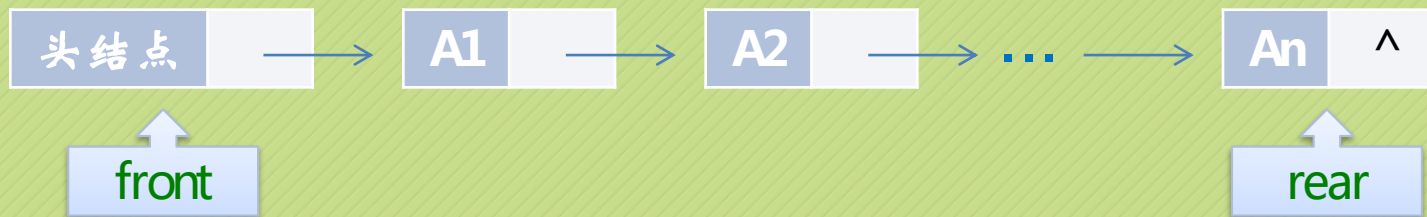
- 队列既可以用链表实现，也可以用顺序表实现。跟栈相反的是，栈一般我们用顺序表来实现，而队列我们常用链表来实现，简称为链队列。

```
typedef struct QNode {  
    ElemType data;  
    struct QNode *next;  
} QNode, *QueuePrt;  
  
typedef struct {  
    QueuePrt front, rear; // 队头、尾指针  
} LinkQueue;
```

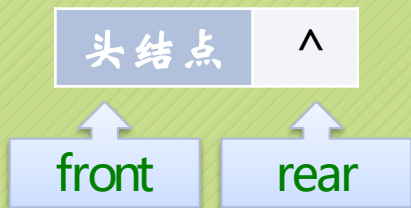


队列的链式存储结构

- 我们将队头指针指向链队列的头结点，而队尾指针指向终端结点。（注：头结点不是必要的，但为了方便操作，我们加上了。）



- 空队列时，front和rear都指向头结点。



创建一个队列

- 创建一个队列要完成两个任务：一是在内存中创建一个头结点，二是将队列的头指针和尾指针都指向这个生成的头结点，因为此时是空队列。

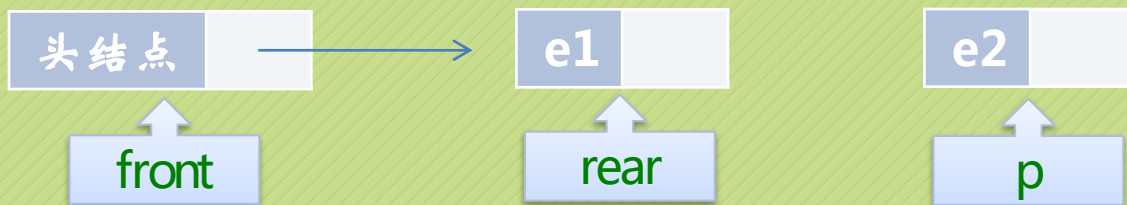
```
initQueue(LinkQueue *q)
{
    q->front=q->rear=(QueuePtr)malloc(sizeof(QNode));
    if( !q->front )
        exit(0);
    q->front->next = NULL;
}
```



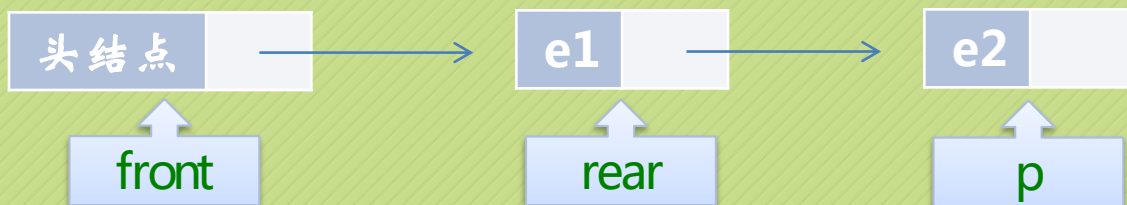
入队列操作

- 入队列的操作过程如下:

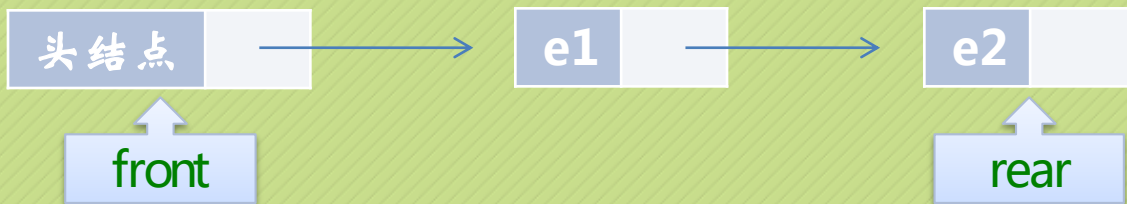
①



②



③



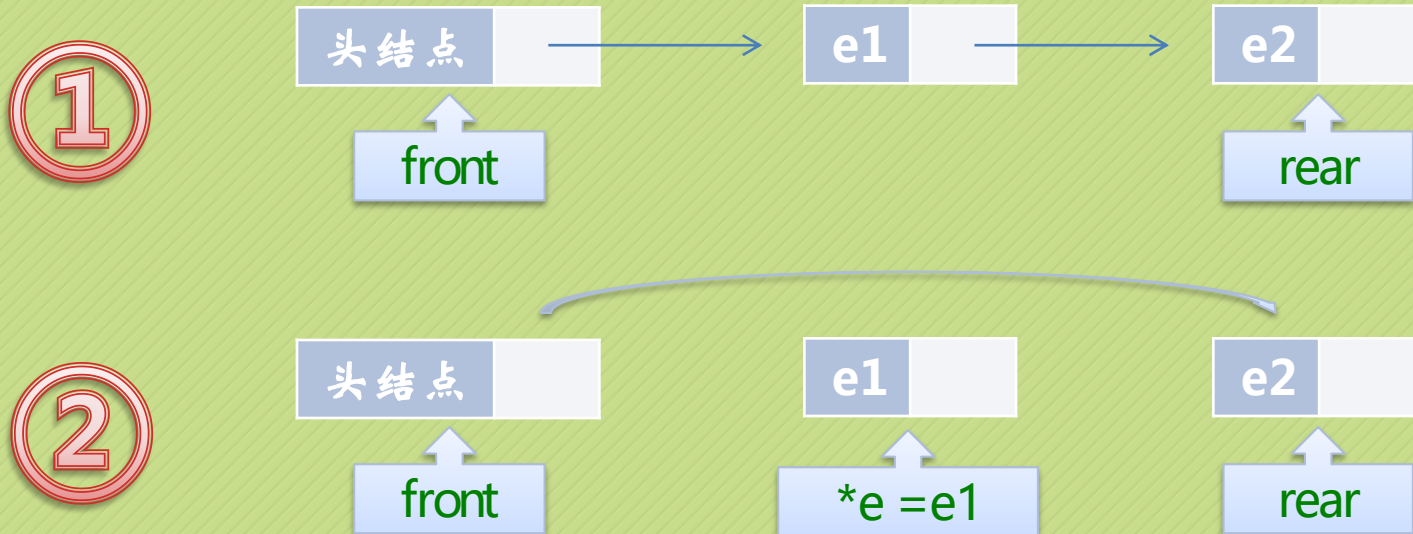
入队列操作

```
InsertQueue(LinkQueue *q, ElemType e)
{
    QueuePtr p;
    p = (QueuePtr)malloc(sizeof(QNode));
    if( p == NULL )
        exit(0);
    p->data = e;
    p->next = NULL;
    q->rear->next = p;
    q->rear = p;
}
```



出队列操作

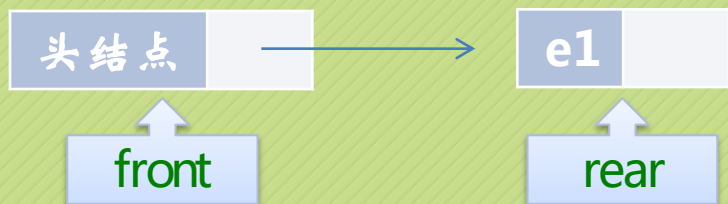
- 出队列操作是将队列中的第一个元素移出，队头指针不发生改变，改变头结点的next指针即可。
- 出队列的操作过程如下：



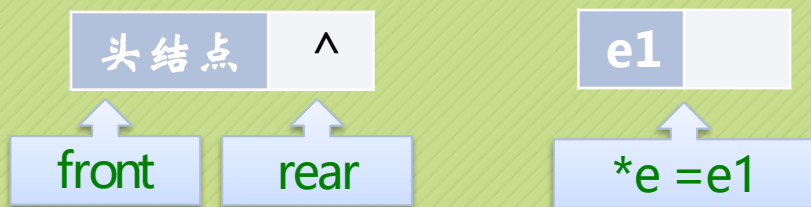
出队列操作

- 如果原队列只有一个元素，那么我们就应该处理一下队尾指针。

①



②



出队列操作

```
DeleteQueue(LinkQueue *q, ELemType *e)
{
    QueuePtr p;
    if( q->front == q->rear )
        return;
    p = q->front->next;
    *e = p->data;
    q->front->next = p->next;
    if( q->rear == p )
        q->rear = q->front;
    free(p);
}
```



销毁一个队列

- 由于链队列建立在内存的动态区，因此当一个队列不再有用时应当把它及时销毁掉，以免过多地占用内存空间。

```
DestroyQueue(LinkQueue *q)
{
    while( q->front ) {
        q->rear = q->front->next;
        free( q->front );
        q->front = q->rear;
    }
}
```



课后作业

- 编写一个链队列，任意输入一串字符，以#作为结束标志，然后将队列中的元素显示到屏幕上。
- 请自行完成后再参考小甲鱼提供的源代码。

