

队列

它是一种特殊的线性表,特殊之处在于它只允许在表的前端进行删除操作,而在表的后端进行插入操作,和栈一样,队列是一种操作受限制的线性表。进行插入操作的端称为队尾,进行删除操作的端称为队头。队列中没有元素时,称为空队列。

队列元素: 队列的数据元素

入队: 在队列中插入一个队列元素

出队: 从队列中删除一个队列元素

因为队列只允许在一端插入,在另一端删除,所以只有最早进入队列的元素才能最先从队列中删除,故队列又称为先进先出(FIFO—first in first out) **线性表**。

顺序队列:

建立顺序队列结构必须为其静态分配或动态申请一片连续的存储空间,并设置两个指针进行管理。一个是队头指针front,它指向队头元素;另一个是队尾指针rear,它指向下一个入队元素的存储位置。

每次在队尾插入一个元素是, rear增1;每次在队头删除一个元素时, front增1。随着插入和删除操作的进行,队列元素的个数不断变化,队列所占的存储空间也在为队列结构所分配的连续空间中移动。当front=rear时,队列中没有任何元素,称为空队列。当rear增加到指向分配的连续空间之外时,队列无法再插入新元素,但这时往往还有大量可用空间未被占用,这些空间是已经出队的队列元素曾经占用过得存储单元。

顺序队列中的栈溢出现象:

“下溢”:当队列为空时,做出队运算产生的溢出现象。是正常现象,常用作程序控制转移的条件。

“真上溢”:当队列满时,做入队运算产生的空间溢出现象。是一种出错的状态,应该设法避免。

“假上溢”:由于入队和出对操作中,头尾指针只增加不减小,致使被删元素的空间永远无法重新利用。当队列中实际的元素个数远远小于向量空间的规模时,也可能由于尾指针已超越向量空间的上界2而不能做入队操作。该现象称为"假上溢"现象。

循环队列:

在实际使用队列时，为了使队列空间能重复使用，往往对队列的使用方法稍加改进：无论插入或删除，一旦rear指针增1或front指针增1时超出了所分配的队列空间，就让它指向这片连续空间的起始位置。自己真从MaxSize-1增1变到0，可用取余运算 $\text{rear} \% \text{MaxSize}$ 和 $\text{front} \% \text{MaxSize}$ 来实现。这实际上是把队列空间想象成一个环形空间，环形空间中的存储单元循环使用，用这种方法管理的队列也就称为循环队列。除了一些简单应用之外，真正实用的队列是循环队列。

在循环队列中，当队列为空时，有 $\text{front} = \text{rear}$ ，而当所有队列空间全占满时，也有 $\text{front} = \text{rear}$ 。为了区别这两种情况，规定循环队列最多只能有MaxSize-1个队列元素，当循环队列中只剩下一个空存储单元时，队列就已经满了。因此，队列判空的条件时 $\text{front} = \text{rear}$ ，而队列判满的条件时 $\text{front} = (\text{rear} + 1) \% \text{MaxSize}$