

等级考试公共基础考点分析之数据结构与算法(6) PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_\\_E7\\_AD\\_89\\_E7\\_BA\\_A7\\_E8\\_80\\_83\\_E8\\_c97\\_133609.htm](https://www.100test.com/kao_ti2020/133/2021_2022__E7_AD_89_E7_BA_A7_E8_80_83_E8_c97_133609.htm) 考点9 顺序表的删除运算 线性表的删除运算是指将表的第 $i(1 \leq i \leq n)$ 个结点删除，使长度为 $n$ 的线性表： $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$  变成长度为 $n-1$ 的线性表  $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$  该算法的时间分析与插入算法相似，结点的移动次数也是由表长 $n$ 和位置 $i$ 决定。若 $i=n$ ，则由于循环变量的初值大于终值，前移语句将不执行，无需移动结点；若 $i=1$ ，则前移语句将循环执行 $n-1$ 次，需移动表中除开始结点外的所有结点。这两种情况下算法的时间复杂度分别为 $O(1)$ 和 $O(n)$ 。删除算法的平均性能分析与插入算法相似。在长度为 $n$ 的线性表中删除一个结点，令 $E_{de}(n)$ 表示所需移动结点的平均次数，删除表中第 $i$ 个结点的移动次数为 $n-i$ ，故式子中， $p_i$ 表示删除表中第 $i$ 个结点的概率。在等概率的假设下， $p_1=p_2=p_3=\dots=p_n=1/n$  由此可得：即在顺序表上做删除运算，平均要移动表中约一半的结点，平均时间复杂度也是 $O(n)$ 。

#### 1.4 栈和队列 考点10 栈及其基本运算

1什么是栈 栈实际也是线性表，只不过是一种特殊的线性表。栈(Stack)是只能在表的一端进行插入和删除运算配线性表，通常称插入、删除的这一端为栈顶(Top)，另一端为栈底(Bottom)。当表中没有元素时称为空栈(栈顶元素总是后被插入的元素，从而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。假设栈 $S=(a_1, a_2, a_3, \dots, a_n)$ ，则 $a_1$ ，称为栈底元素， $a_n$ 为栈顶元素。栈中元素按 $a_1, a_2, a_3, \dots, a_n$ 的次序进栈，退栈的第

一个元素应为栈顶元素。换句话说，栈的修改是按后进先出的原则进行的。因此，栈称为先进后出表(FILO, First In Last Out)，或“后进先出”表(LIFO, Last In First Out)，如图1-7所示。

2. 栈的顺序存储及其运算

(1) 入栈运算：入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加一(即top加1)，然后将元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的最后一个位置时，说明栈空间已满，不可能再进行入栈操作。这种情况称为栈“上溢”错误。如图1-8所示。

(2) 退栈运算：退栈是指取出栈顶元素并赋给一个指定的变量。首先将栈顶元素(栈顶指针指向的元素)赋给一个指定的变量，然后将栈顶指针减一(即top减1)。当栈顶指针为0时，说明栈空，不可进行退栈操作。这种情况称为栈的“下溢”错误。

(3) 读栈顶元素：读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素，只是将它赋给一个变量，因此栈顶指针不会改变。当栈顶指针为0时，说明栈空，读不到栈顶元素。

考点11 队列及其基本运算

1. 什么是队列

队列(queue)是只允许在一端删除，在另一端插入的顺序表，允许删除的一端叫做队头(front)，允许插入的一端叫做队尾(rear)，当队列中没有元素时称为空队列。在空队列中依次加入元素 $a_1, a_2, \dots, a_n$ 之后， $a_1$ 是队头元素， $a_n$ 是队尾元素。显然退出队列的次序也只能是 $a_1, a_2, \dots, a_n$ 也就是说队列的修改是依先进先出的原则进行的。因此队列亦称作先进先出(FIFO, First In First Out)的线性表，或后进后出(LILO, Last In Last Out)的线性表。往队列队尾插入一个元素称为入队运算，从队列的排头删除一个元素称为退队运算，如图1-10所示。

一个队列的。删除个儿素后的队列间插入元素E

后的队列 2 循环队列及其运算 在实际应用中，队列的顺序存储结构一般采用循环队列的形式。所谓循环队列，就是将队列存储空间最后一个位置绕到第一个位置，形成逻辑上的环状空间。在循环队列中，用队尾指针 rear 指向队列中的队尾元素，用排头指针 front 指向排头元素的前一个位置。因此，从排头指针 front 指向的后一个位置直到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素。可以将向量空间想象为一个首尾相接的圆环，如图 1-12 所示，并称这种向量为循环向量，存储在其中的队列称为循环队列 (Circular Queue)。在循环队列中进行出队、入队操作时，头尾指针仍要加 1，朝前移动。只不过当头尾指针指向向量上界 (QueueSize-1) 时，其加 1 操作的结果是指向向量的下界 0。由于入队时尾指针向前追赶头指针，出队时头指针向前追赶尾指针，故队空和队满时头尾指针均相等。因此，我们无法通过 front=rear 来判断队列“空”还是“满”。在实际使用循环队列时，为了能区分队列满还是队列空，通常还需增加一个标志、s、值的定义如下：当 s=0 时表示队列空；当 s=1 时表示队列非空。

(I) 入队运算 入队运算是指在循环队列的队尾加入一个新元素。首先将队尾指针进一 (即 rear=rear+1)，并当 rear=QueueSize-1 时置 rear=0；然后将新元素插入到队尾指针指向的位置。当循环队列非空 (s=1) 且队尾指针等于队头指针时，说明循环队列已满，不能进行入队运算，这种情况称为“上溢”。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)