

2009年自考《数据结构》各章要点自考 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/532/2021\\_2022\\_2009\\_E5\\_B9\\_B4\\_E8\\_87\\_AA\\_c67\\_532580.htm](https://www.100test.com/kao_ti2020/532/2021_2022_2009_E5_B9_B4_E8_87_AA_c67_532580.htm) 第一章 概论 数据就是指能够被计算机识别、存储和加工处理的信息的载体。数据元素是数据的基本单位，可以由若干个数据项组成。数据项是具有独立含义的最小标识单位。数据结构的定义：逻辑结构：从逻辑结构上描述数据，独立于计算机。线性结构：一对一关系。线性结构：多对多关系。存储结构：是逻辑结构用计算机语言的实现。顺序存储结构：如数组。链式存储结构：如链表。稠密索引：每个结点都有索引项。稀疏索引：每组结点都有索引项。散列存储结构：如散列表。对数据的操作：定义在逻辑结构上，每种逻辑结构都有一个运算集合。常用的有：检索、插入、删除、更新、排序。数据类型：是一个值的集合以及在这些值上定义的一组操作的总称。原子类型：由语言提供。结构类型：由用户借助于描述机制定义，是导出类型。抽象数据类型ADT：是抽象数据的组织和与之的操作。相当于在概念层上描述问题。优点是将数据和操作封装在一起实现了信息隐藏。程序设计的实质是对实际问题选择一种好的数据结构，设计一个好的算法。算法取决于数据结构。算法是一个良定义的计算过程，以一个或多个值输入，并以一个或多个值输出。评价算法的好坏的因素：算法是正确的；执行算法的时间；执行算法的存储空间(主要是辅助存储空间)；算法易于理解、编码、调试。时间复杂度：是某个算法的时间耗费，它是该算法所求解问题规模 $n$ 的函数。|百考试题收集| 渐近时间复杂度：是指当问题规模趋向无

穷大时，该算法时间复杂度的数量级。评价一个算法的时间性能时，主要标准就是算法的渐近时间复杂度。算法中语句的频度不仅与问题规模有关，还与输入实例中各元素的取值相关。时间复杂度按数量级递增排列依次为：常数阶 $O(1)$ 、对数阶 $O(\log_2 n)$ 、线性阶 $O(n)$ 、线性对数阶 $O(n \log_2 n)$ 、平方阶 $O(n^2)$ 、立方阶 $O(n^3)$ 、…… $k$ 次方阶 $O(n^k)$ 、指数阶 $O(2^n)$ 。空间复杂度：是某个算法的空间耗费，它是该算法所求解问题规模 $n$ 的函数。算法的时间复杂度和空间复杂度合称算法复杂度。

## 第二章 线性表

线性表是由 $n \geq 0$ 个数据元素组成的有限序列。 $n=0$ 是空表.非空表，只能有一个开始结点，有且只能有一个终端结点。线性表上定义的基本运算：

构造空表：Initlist(L) 求表长：Listlength(L) 取结点：  
GetNode(L, i) 查找：LocateNode(L, x) 插入：InsertList(L, x, i) 删除：Delete(L, i)

顺序表是按线性表的逻辑结构次序依次存放在一组地址连续的存储单元中。在存储单元中的各元素的物理位置和逻辑结构中各结点相邻关系是一致的。地址计算： $LOCa(i) = LOCa(1) + (i-1) * d$  (首地址为1) / 考试大收集整理 / 在顺序表中实现的基本运算：插入：平均移动结点次数为 $n/2$ .平均时间复杂度均为 $O(n)$ 。删除：平均移动结点次数为 $(n-1)/2$ .平均时间复杂度均为 $O(n)$ 。

线性表的链式存储结构中结点的逻辑次序和物理次序不一定相同，为了能正确表示结点间的逻辑关系，在存储每个结点值的同时，还存储了其后续结点的地址信息(即指针或链)。这两部分信息组成链表中的结点结构。一个单链表由头指针的名字来命名。单链表运算：建立单链表 头插法： $s \rightarrow next = s, r = s$ . 平均时间复杂度均为 $O(n)$  加头结点的算法：对开始结点的操作无需特殊处

理，统一了空表和非空表。查找按序号：与查找位置有关，平均时间复杂度均为 $O(n)$ 。按值：与输入实例有关，平均时间复杂度均为 $O(n)$ 。插入运算： $p = \text{GetNode}(L, i-1).s \rightarrow \text{next} \rightarrow p \rightarrow \text{next} \rightarrow p \rightarrow \text{next} \rightarrow \text{free}(r)$ 。平均时间复杂度均为 $O(n)$ 。单循环链表是一种首尾相接的单链表，终端结点的指针域指向开始结点或头结点。链表终止条件是以指针等于头指针或尾指针。采用单循环链表在实用中多采用尾指针表示单循环链表。优点是查找头指针和尾指针的时间都是 $O(1)$ ，不用遍历整个链表。双链表就是双向链表，就是在单链表的每个结点里再增加一个指向其直接前趋的指针域prior，形成两条不同方向的链。由头指针head惟一确定。双链表也可以头尾相链接构成双(向)循环链表。双链表上的插入和删除时间复杂度均为 $O(1)$ 。顺序表和链表的比较：基于空间：顺序表的存储空间是静态分配，存储密度为1；适于线性表事先确定其大小时采用。链表的存储空间是动态分配，存储密度 $\leq 1$ ；适于线性表长度变化大时采用。基于时间：顺序表是随机存储结构，当线性表的操作主要是查找时，宜采用。以插入和删除操作为主的线性表宜采用链表做存储结构。若插入和删除主要发生在表的首尾两端，则宜采用尾指针表示的单循环链表。

### 第三章 栈和队列

栈(Stack)是仅限制在表的一端进行插入和删除运算的线性表，称插入、删除这一端为栈顶，另一端称为栈底。表中无元素时为栈空。栈的修改是按后进先出的原则进行的，我们又称栈为LIFO表(Last In First Out)。通常栈有顺序栈和链栈两种存储结构。栈的基本运算有六种：构造空栈： $\text{InitStack}(S)$  判栈空： $\text{StackEmpty}(S)$  判栈满： $\text{StackFull}(S)$  进栈： $\text{Push}(S, x)$  退栈： $\text{Pop}(S)$  取栈顶元素

: StackTop(S) 在顺序栈中有“上溢”和“下溢”的现象。“上溢”是栈顶指针指出栈的外面是出错状态。“下溢”可以表示栈为空栈，因此用来作为控制转移的条件。顺序栈中的基本操作有六种：构造空栈 判栈空 判栈满 进栈 退栈 取栈顶元素 链栈则没有上溢的限制，因此进栈不要判栈满。链栈不需要在头部附加头结点，只要有链表的头指针就可以了。链栈中的基本操作有五种：构造空栈 判栈空 进栈 退栈 取栈顶元素 队列(Queue)是一种运算受限的线性表，插入在表的一端进行，而删除在表的另一端进行，允许删除的一端称为队头(front)，允许插入的一端称为队尾(rear)，队列的操作原则是先进先出的，又称作FIFO表(First In First Out)。队列也有顺序存储和链式存储两种存储结构。考试大整理 队列的基本运算有六种：置空队：InitQueue(Q) 判队空：QueueEmpty(Q) 判队满：QueueFull(Q) 入队：EnQueue(Q, x) 出队

: DeQueue(Q) 取队头元素：QueueFront(Q) 顺序队列的“假上溢”现象：由于头尾指针不断前移，超出向量空间。这时整个向量空间及队列是空的却产生了“上溢”现象。为了克服“假上溢”现象引入循环向量的概念，是把向量空间形成一个头尾相接的环形，这时队列称循环队列。判定循环队列是空还是满，方法有三种：一种是另设一个布尔变量来判断；第二种是少用一个元素空间，入队时先测试 $((rear + 1) \% m = front)$ ? 满：空；第三种就是用一个计数器记录队列中的元素的总数。队列的链式存储结构称为链队列，一个链队列就是一个操作受限的单链表。为了便于在表尾进行插入(入队)的操作，在表尾增加一个尾指针，一个链队列就由一个头指针和一个尾指针唯一地确定。链队列不存在队满和上溢的问题

。在链队列的出队算法中，要注意当原队中只有一个结点时，出队后要同进修改头尾指针并使队列变空。

#### 第四章 串

串是零个或多个字符组成的有限序列。空串：是指长度为零的串，也就是串中不包含任何字符(结点)。空白串：指串中包含一个或多个空格字符的串。在一个串中任意个连续字符组成的子序列称为该串的子串，包含子串的串就称为主串。子串在主串中的序号就是指子串在主串中首次出现的位置。空串是任意串的子串，任意串是自身的子串。串分为两种：串常量在程序中只能引用不能改变；串变量的值可以改变。串的基本运算有：求串长`strlen(char*s)` 串复制`strcpy(char*to, char*from)` 串联接`strcat(char*to, char*from)` 串比较`strcmp(char*s1, char*s2)` 字符定位`strchr(char*s, charc)`。

串是特殊的线性表(结点是字符)，所以串的存储结构与线性表的存储结构类似。串的顺序存储结构简称为顺序串。顺序串又可按存储分配的不同分为：静态存储分配：直接用定长的字符数组来定义。优点是涉及串长的操作速度快，但不适合插入、链接操作。动态存储分配：是在定义串时不分配存储空间，需要使用时按所需串的长度分配存储单元。串的链式存储就是用单链表的方式存储串值，串的这种链式存储结构简称为链串。链串与单链表的差异只是它的结点数据域为单个字符。为了解决“存储密度”低的状况，可以让一个结点存储多个字符，即结点的大小。

#### 顺序串上子串定位的运算

：又称串的“模式匹配”或“串匹配”，是在主串中查找出子串出现的位置。在串匹配中，将主串称为目标(串)，子串称为模式(串)。这是比较容易理解的，串匹配问题就是找出给定模式串P在给定目标串T中首次出现的有效位移或者是全

部有效位移。最坏的情况下时间复杂度是 $O((n-m+1)m)$ ，假如 $m$ 与 $n$ 同阶的话则它是 $O(n^2)$ 。链串上的子串定位运算位移是结点地址而不是整数。

\考试大收集\第五章 多维数组和广义表

数组一般用顺序存储的方式表示。存储的方式有：行优先顺序，也就是把数组逐行依次排列。PASCAL、C 列优先顺序，就是把数组逐列依次排列。FORTRAN 地址的计算方法：

：按行优先顺序排列的数组： $LOCa(ij)=LOCa(11) ((i-1)*n+(j-1))*d$ . 按列优先顺序排列的数组： $LOCa(ij)=LOCa(11) ((j-1)*n+(i-1))*d$ . 矩阵的压缩存储：为多个相同的非零元素分配一个存储空间.对零元素不分配空间。特殊矩阵的概念：所谓特殊矩阵是指非零元素或零元素分布有一定规律的矩阵。

稀疏矩阵的概念：一个矩阵中若其非零元素的个数远远小于零元素的个数，则该矩阵称为稀疏矩阵。特殊矩阵的类型：

对称矩阵：满足 $a(ij)=a(ji)$ 。元素总数 $n(n+1)/2$ .  $I=\max(i, j)$ ,  $J=\min(i, j)$ ,  $LOCa(ij)=LOC(sa[0]) (I*(I+1)/2+J)*d$ . 三角矩阵：

：上三角阵： $k=i*(2n-i+1)/2+j-i$ ,  $LOCa(ij)=LOC(sa[0]) k*d$ . 下三角阵： $k=i*(i+1)/2+j$ ,  $LOCa(ij)=LOC(sa[0]) k*d$ . 对角矩阵： $k=2ij$ ,  $LOCa(ij)=LOC(sa[0]) k*d$ . 稀疏矩阵的压缩存储方式用三元组表把非零元素的值和它所在的行号列号做为一个结点存放在一起，用这些结点组成的一个线性表来表示。但这种压缩存储方式将失去随机存储功能。加入行表记录每行的非零元素在三元组表中的起始位置，即带行表的三元组表。

广义表是 $n(n \geq 0)$ 个元素的有限序列，其中的元素是原子或者是一个广义表。广义表表头和表尾的概念：若广义表 $LS$ 非空( $n \geq 1$ )，则这个广义表的第一个元素就是表头。其余的元素组成的表称为 $LS$ 的表尾，所以表尾必是一个子表。广义表

有两种表示法，一种是括号表示法，一种是图形表示法。广义表与树(形结构)相对应，这个广义表就是纯表。如果一个广义表的结点又可以被其他结点所共享，则这个表称为再入表。允许递归的表称为递归表。线性表 纯表(树) 再入表 递归表。可见，广义表是对线性表和树的推广。广义表有两个特殊的基本运算：取表头head(LS)：取表中的第一个数据元素，不能对空表操作。取表尾tail(LS).取除表头外，其余数据元素构成的子表，不能对空表操作。百考试题收集 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)