

全国计算机应用软件人员水平考试辅导丛书

高级程序员 水平考试指导

郑人杰 主编

清华大学出版社



全国计算机应用软件人员水平考试辅导丛书

高级程序员 水平考试指导

郑人杰 主编

清华大学出版社

内 容 简 介

本书为计算机软件人员水平考试（高级程序员级）的复习辅导教材，它是在中国软件行业协会考试指导中心组织下，由北京大学、清华大学等院校的有关专家在原编写出版的《软件人员水平考试辅导》丛书（杨德元主编）基础上，依照1990年公布的《中国计算机应用软件人员水平考试大纲》重新改编而成的。内容包括：软件知识、硬件知识和综合基础知识。各章均有一定数量的例题、习题，以指导软件人员参加水平考试。

本书可供计算机软件人员应考前复习、提高用，也可供大专院校师生和工程技术人员学习参考。可供各部门举办辅导班用作教材（配有录像带）。本书也是中央电视台电视辅导讲座教材。

JS442/3/03

全国计算机应用软件人员水平考试辅导丛书

高级程序员水平考试指导

郑人杰 主编

☆

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

☆

开本：787×1092 1/16 印张：28 字数：664 千字

1991年3月第1版 1991年3月第1次印刷

印数：0001~20000

ISBN 7-302-00846-9/TP·306

定价：11.50 元

《高级程序员水平考试指导》

编 委 会

顾问: 杨天行 徐家福 杨芙清 王尔乾

主编: 郑人杰

编委: (按姓氏笔划为序)

丁文魁	方 裕	王爱英	卢义明	史美林	刘椿年
田金兰	朱慧真	何克忠	沈林兴	严俭和	杨冬青
陈向群	屈婉玲	罗晓沛	俞盘祥	柳西玲	相士俊
徐国平	殷志鹤	耿素云	潘锦平		

前 言

计算机软件人员水平考试是造就宏大的多层次计算机人才队伍的一项重要措施,从1990年起,在全国实施《中国计算机应用软件人员专业技术职务任职资格(水平)考试暂行规定》,资格考试级别分为程序员级(相当于技术员、助理工程师)、高级程序员级(相当于工程师)、系统分析员级(相当于高级工程师)。

为了指导软件人员的学习,帮助准备参加软件水平考试的学员应试,我们在原编写出版的《全国计算机应用软件人员水平考试辅导丛书》(杨德元主编,1987—1989)基础上,重新聘请了北京大学、清华大学、中国科大研究生院、上海交通大学、南京大学、南京东南大学等单位的专家组成编委会,根据1990年2月公布的考试大纲,进一步改编成本丛书。

本丛书共包括:《程序员水平考试指导》、《高级程序员水平考试指导》、《程序设计考试指导(程序员级、高级程序员级)》、《软件人员水平考试题例和选解》及《系统分析员教程》。

必须说明,本丛书是供水平考试应考人员参考的综合性辅导教材,供读者在自学基础知识后复习提高用。它的主要特点是全面(覆盖了考试大纲的主要内容)、系统(而不是给出零碎的知识),并且体现了基本要求(包括基本概念、基本方法和基本原理)。限于篇幅,各章节不可能全面展开。为解决读者自学问题,我们将另行组织编写《全国计算机软件人员水平考试系列教材》丛书。

本丛书亦可用作在职软件人员进修教材,或计算机应用专业师生的参考书籍,本丛书也是中央电视台播放的计算机软件人员水平考试电视辅导讲座的教材。

本书供高级程序员级使用,共分为三个部分,即:软件知识、硬件知识和综合基础知识,本书由郑人杰主编,参加编写的有:北京大学耿素云、屈婉玲(离散数学)、杨冬青(数据结构)、丁文魁(语言处理程序)、方裕(文件系统和操作系统)、陈向群(汉字信息处理);清华大学严俭和(程序语言)、田金兰(实用程序)、俞盘祥(数据库系统)、相士俊(管理信息系统)、郑人杰(软件工程)、王爱英、卢义明(硬件部分)、何克忠(实时处理)、柳西玲(计算机辅助处理系统)和中国科大研究生院罗晓沛(数值计算)等,徐国平参加了编审。

在本丛书的编写过程中,曾得到中国计算机学会的指导和帮助。清华大学出版社为使本丛书迅速出版付出了辛勤劳动,在此一并致谢!

中国软件行业协会考试指导中心

1990年11月

目 录

第一部分 软件知识

第一章 数据结构..... 2	5.2 进程及处理机管理.....95
1.1 数据结构基础知识..... 2	5.3 存储管理..... 100
1.2 线性表..... 4	5.4 设备管理..... 105
1.3 树与二叉树..... 8	5.5 作业管理..... 106
1.4 图.....12	5.6 其它管理..... 109
1.5 多维数组、稀疏矩阵和 广义表.....17	5.7 实例分析..... 112
1.6 集合.....20	第六章 实用程序..... 118
1.7 查找.....23	6.1 编辑程序的功能..... 118
1.8 排序.....29	6.2 连接装入程序的功能..... 120
1.9 题例分析.....35	6.3 调试程序的功能..... 120
1.10 习题..... 43	6.4 格式加工程序的功能..... 122
参考书.....48	第七章 数据库系统..... 124
第二章 高级语言综述.....49	7.1 数据管理方法的发展 及基本概念..... 124
2.1 程序设计语言概述.....49	7.2 三种数据模型..... 128
2.2 变量和表达式.....54	7.3 典型的数据库系统..... 130
2.3 数据类型.....57	7.4 关系数据库设计理论..... 138
2.4 语句.....61	7.5 数据库保护..... 142
2.5 子程序.....66	参考书..... 143
2.6 程序结构与模块.....69	第八章 数据通信和计算机网络..... 144
2.7 习题.....71	8.1 数据通信基础..... 144
第三章 语言处理程序.....74	8.2 计算机网络基础..... 154
3.1 概述.....74	8.3 题例分析..... 167
3.2 编译技术.....76	8.4 Ethernet 和 Novellnpt 简介..... 168
3.3 习题.....82	参考书..... 176
参考书.....83	第九章 软件工程..... 177
第四章 文件系统.....84	9.1 软件工程概述..... 177
4.1 概述.....84	9.2 软件开发计划的制定..... 180
4.2 文件控制块与文件目录.....85	9.3 软件需求分析..... 181
4.3 文件的分类.....87	9.4 软件设计..... 186
4.4 文件的组织.....88	9.5 结构化程序设计与程序 设计风格..... 197
4.5 文件的使用.....90	9.6 软件测试..... 202
第五章 操作系统.....92	9.7 软件维护..... 205
5.1 概述.....92	

9.8 软件工程的文档编制.....	211	9.10 软件管理.....	220
9.9 软件工程标准化.....	216		

第二部分 硬件知识

第十章 计算机中数据的表示、转换和运算	229	13.4 中断系统.....	275
10.1 数制及其转换.....	229	13.5 输入输出接口.....	278
10.2 数据的表示及其运算.....	230	13.6 汉字输入输出.....	281
10.3 逻辑变量及逻辑运算.....	237	13.7 联机、脱机和假脱机.....	282
10.4 习题.....	239	13.8 磁带机中使用的纠错码CRC.....	283
第十一章 计算机的组成	240	13.9 计算机网络.....	285
11.1 计算机的基本组成.....	240	13.10 习题.....	287
11.2 指令系统.....	242	第十四章 安全性、可靠性和系统性能评价初步	288
11.3 中央处理机.....	247	14.1 计算机的可靠性.....	288
11.4 控制器的实现原理.....	251	14.2 计算机的故障、诊断和容错技术.....	290
11.5 练习题.....	252	14.3 数据安全和保密.....	292
第十二章 存储器系统	254	14.4 计算机的性能测试和性能评价.....	293
12.1 存储体系的形成.....	254	14.5 模拟与仿真.....	295
12.2 主存储器.....	256	14.6 习题.....	296
12.3 虚拟存储器.....	259	第十五章 计算机体系结构的发展	298
12.4 高速缓冲存储器 Cache.....	262	15.1 流水线组织.....	298
12.5 存储保护.....	264	15.2 并行处理概念.....	301
12.6 习题.....	266	15.3 多机系统.....	302
第十三章 输入输出系统	268	15.4 精简指令系统计算机 RISC	304
13.1 输入输出设备.....	268	15.5 习题.....	311
13.2 辅助存储器.....	270	参考书.....	314
13.3 输入输出系统的工作方式.....	273		

第三部分 综合基础知识

第十六章 离散数学	314	17.4 数值微分与数值积分.....	352
16.1 数理逻辑.....	314	17.5 线性代数方程组的数值解法.....	354
16.2 集合论.....	321	17.6 常微分方程数值解法.....	363
16.3 代数结构.....	326	17.7 习题.....	371
16.4 图论.....	329	参考书.....	372
16.5 组合分析简介.....	333	第十八章 管理信息处理	373
16.6 题例分析.....	335	18.1 管理信息系统.....	373
16.7 习题.....	337	18.2 经济管理与企业管理知识.....	376
参考书.....	340	18.3 管理科学——运筹学的应用.....	379
第十七章 数值计算	341	18.4 信息系统开发步骤及表达工具.....	381
17.1 算法与误差.....	341	第十九章 实时处理	385
17.2 函数插值与最小二乘法.....	342	19.1 实时系统.....	385
17.3 方程求根.....	350	习题 1.....	389

19.2 数据的采集和处理.....	389	20.6 三维图形显示原理.....	418
习题 2.....	395	20.7 CAD 系统设计原理.....	419
19.3 自动化技术.....	395	20.8 计算机辅助教学.....	421
习题 3.....	403	参考书.....	423
参考书.....	404	第二十一章 汉字信息处理.....	424
第二十章 计算机辅助系统.....	405	21.1 汉字键盘输入方案评测规则.....	424
20.1 CAD 系统的硬软件支持.....	405	21.2 GB 2311-80《信息处理交换用七 位编码字符集的扩充方法》.....	426
20.2 CAD 系统的分类及特点.....	406	21.3 信息交换用汉字编码辅助集.....	428
20.3 CAD 系统设计中的基本概念.....	407	参考书.....	429
20.4 交互式图形显示的基本技术.....	408	附录 部分习题答案.....	430
20.5 卡通技术.....	418		

第一部分 软件知识

计算机软件是计算机系统的灵魂。高级程序员熟练地掌握软件知识对他们参加软件开发项目、使用外购的软件以及维护已投入运行的软件都是非常必要的。

通常认为,计算机软件包括系统软件、支撑软件和应用软件三个方面。对于一个应用软件人员来说,虽然日常面临的工作任务大都属于自己业务领域内的应用软件问题,仍然必须对系统软件的有关知识有较为深入的理解。初步掌握了程序语言和数据结构的基本知识,也确实可以编写小型的程序,但进一步接触软件课题便会涉及到许多系统软件的知识。

本书第一部分以介绍系统软件为主,分为九章。第一章是在掌握程序员级要求的数据结构基本知识基础上,系统讲述了线性表、树、图、多维数组、集合、查找、排序等专题。第二章对若干高级语言进行了综述。它是假定读者已经熟悉了一至两种高级语言的使用,为了能够使读者对程序语言的一般特性有更为深入的理解,并且与语言的实现建立联系,这一章对程序语言的特性作了分析、比较和横向归纳。第三章讲的是语言实现问题,即所谓“编译技术”。其中包括汇编、编译和解释系统的基本原理。第四和第五章的文件系统和操作系统分别就文件管理和进程及处理机、存储、设备、作业的管理问题给出了系统描述。第六章的实用程序包括了最为常见的编辑、连接装配、调试和格式加工程序的功能。第七章在介绍数据库基本概念的基础上,给出了目前比较流行的三种数据模型和典型的 dBASE 系统。并且对关系数据库理论和数据库保护作了简要的描述。第八章介绍数据通信与计算机网络基础,并介绍了 Ethernet 与 Novell 两种目前最为流行的微机网络。第九章则以软件生存期为线索,比较系统地介绍了软件开发和维护的有关概念;结合实用,这一章中还讨论了软件文档编制和软件工程标准的制定与推行中的有关问题。

第一章 数据结构

计算机科学的各领域及有关的应用软件都要用到各种数据结构。语言编译要使用栈、散列表及语法树；操作系统中用队列、存储管理表及目录树等；数据库系统运用散列表、多链表及索引树等进行数据管理；而在人工智能领域依求解问题性质的差异将涉及到各种不同的数据结构，如广义表、集合、搜索树及各种有向图等等。本章将介绍数据结构的基本概念和一些常用的数据结构，阐明数据结构内在的逻辑关系，讨论它们在计算机中的存储表示，并结合各种典型应用说明它们在进行各种运算时的动态性质及实际的执行算法。详细内容请参阅有关教科书。

1.1 数据结构基础知识

1.1.1 基本概念及术语

1. 数据与数据结构

数据是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序加工处理的符号集合。

数据元素是数据的基本单位，即数据集合中的个体。有时一个数据元素由若干个数据项组成，在这种情况下，称数据元素为记录。数据项是数据的最小单位。而由记录所组成的线性表为文件。

数据对象是具有相同特性的数据元素的集合，是数据的子集。

被计算机加工的数据元素不是孤立无关的，它们彼此间一般存在着某些联系。通常将数据元素间的这种联系关系称为结构。

数据结构就是带有结构特性的数据元素的集合。

2. 数据的逻辑结构和存储结构

只抽象地反映数据元素集合的结构，而不管其存储方式的数据结构称为数据的逻辑结构；相应地，把逻辑结构映象到计算机中所得到的存储方式称为数据的存储结构，亦称为物理结构。

数据结构在计算机中的映象（即表示）包括数据元素和关系的映象两部分，而把表征某一数据结构特点及关联方式的基本信息单位称为该数据结构的结点。结点往往又分成几个域，每个域用来存放与这个结点有关的数据项的信息。以学生登记表为例，可直观地搞清这些概念。如表 1.1，对于每个学生，所登记的信息共 4 项：姓名、学号、所学专业及下一个学生的登记信息位置（存储地址）。

表 1.1 学生登记表

位置	姓 名 NAME	专 业 SPE	学 号 NO	指 针 LINK
2000	John	MATH	806211	2013
2013	Mary	COMPUTE	808209	2026
2026	Kent	RADAR	807115	2078
2039	Jack	RADIO	801322	2052
2052	Jane	PHYS	801124	2065
2065	Barber	COMPUTE	808214	^
2078	Blake	RADIO	801215	2039

这里把每个学生有关的信息集合作为一个整体，看作一个结点。这种结点由 4 个域组成（姓名，专业，学号，指针），其结构形式为：

NAME	SPE	NO	LINK
------	-----	----	------

普遍地，数据结构中结点的域按其性质可分成两大类，一类是存放结点本身的信息域，如上面例子中的 NAME, NO, SPE 域，通常称之为信息域，可用标识符 INFO 表示这些域的全体；另一类域用来存放这一结点与其它结点的连接信息，如上面例子中的 LINK 域。连接信息域的值可能是指向其它结点的一个或多个指针，也可能是其它形式的连接信息。总之，按域的职能和性质来分，一个结点是由自身信息域和连接信息域组成的，其一般结构为：

INFO	LINK
------	------

这里的结点结构是指一般情况而言。在某些存储结构中，结点可以不包括连接信息域，这时的结点结构呈这种形式：

INFO

3. 数据的运算

数据的运算是定义在数据的逻辑结构上的，但运算的具体实现要在存储结构上进行。数据的各种逻辑结构有相应的各种运算，每种逻辑结构都有一个运算的集合。常用的运算有检索、插入、删除、更新、排序等。

数据的运算是数据结构的一个重要方面。讨论任何一种数据结构时都离不开对该结构上的数据运算及实现算法的讨论。

1.1.2 数据的存储结构

数据的存储结构是逻辑结构在计算机存储器里的实现。下面介绍两类最主要的存储结构。大多数数据结构的存储映象都采用其中一类方式，或两类方式的结合。

1. 顺序存储结构

这种存储方式主要用于线性的数据结构，它把逻辑上相邻的数据元素存储在物理上相邻的存储单元里，结点之间的关系由存储单元的邻接关系来体现。

顺序存储结构的主要特点是：①结点中只有自身信息域，没有连接信息域，因此存储密度大，存储空间利用率高。②可以通过计算直接确定数据结构中第 i 个结点的存储地址 L_i ，计算公式为： $L_i = L_0 + (i - 1) * m$ ，其中 L_0 为第一个结点的存储地址，

m 为每个结点所占用的存储单元个数。③插入, 删除运算不便, 会引起大量结点的移动。这一点在下一节还会具体讲到。

2. 链式存储结构

链式存储结构就是在每个结点中至少包括一个指针域, 用指针来体现数据元素之间逻辑上的联系。这种存储结构可把人们从计算机存储单元的相继性限制中解放出来。可以把逻辑上相邻的两个元素存放在物理上不相邻的存储单元中; 还可以在线性编址的计算机存储器中表示结点之间的非线性联系。

链式存储结构的主要特点是: ① 结点中除自身信息外, 还有表示连接信息的指针域, 因此比顺序存储结构的存储密度小, 存储空间利用率低。②逻辑上相邻的结点物理上不必邻接, 可用于线性表、树、图等多种逻辑结构的存储表示。③插入、删除操作灵活方便, 不必移动结点, 只要改变结点中的指针值即可。这一点在下一节还会具体讲到。

1.2 线性表

线性表是最简单、最常用的一种数据结构。线性表的逻辑结构是 n 个数据元素的有限序列 $(a_1, a_2, \dots, a_n)$ 。

用顺序存储结构存储的线性表称作顺序表。

用链式存储结构存储的线性表称作链表。

对线性表的插入、删除运算可以发生的位置加以限制, 则是两种特殊的线性表——栈和队列。

1.2.1 顺序表和一维数组

各种高级语言里的一维数组就是用顺序方式存储的线性表, 因此也常用一维数组来称呼顺序表。

下面主要讨论顺序表的插入和删除运算。

往顺序表中插入一个新结点时, 可能要移动一系列结点。若顺序表中结点个数为 n , 在往每个位置插入的概率相等的情况下, 插入一个结点平均需要移动的结点个数为 $n/2$ 。

类似地, 从顺序表中删除一个结点可能需要往前移动一系列结点。在等概率的情况下, 删除一个结点平均需要移动的结点个数也是 $n/2$ 。

1.2.2 链表

1. 线性链表 (单链表)

线性链表就是链式存储的线性表, 其结点中只含有一个指针域, 用来指出其后继结点的位置。线性链表的最后一个结点没有后继结点, 它的指针域为空 (记为 NIL 或 $\wedge$)。另外还需要设置一个指针 $head$, 指向线性链表的第一个结点。

链表的一个重要特点是插入、删除运算灵活方便, 不需移动结点, 只要改变结点中指针域的值即可。

图 1.1 显示了在单链表中指针 P 所指结点后插入一个新结点的指针变化情况，虚线所示为变化后的指针。

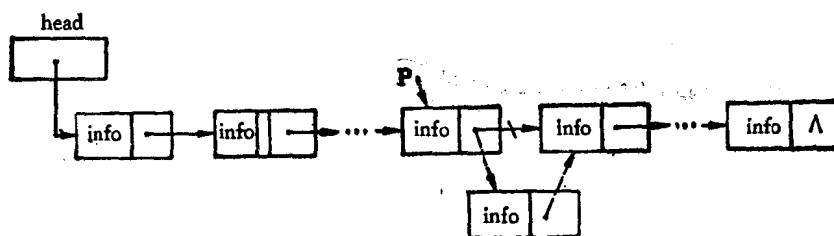


图 1.1 单链表的插入

图 1.2 显示了从单链表中删除指针 P 所指结点的下一个结点的指针变化情况，虚线所示为变化后的指针。

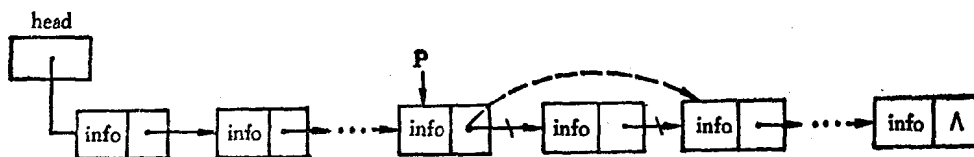


图 1.2 单链表的删除

在线性链表中，往第一个结点前面插入新结点和删除第一个结点会引起表头指针 head 值的变化。通常可以在线性链表的第一个结点之前附设一个结点，称为头结点。头结点的数据域可以不存储任何信息。头结点的指针域存储指向第一个结点的指针，如图 1.3 所示。这样，往第一个结点前面插入新结点和删除第一个结点就不影响表头指针 head 的值，而只改变结点的指针域的值，因此就可以和其它位置的插入、删除同样处理了。

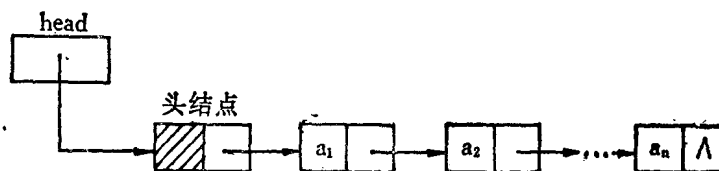


图 1.3 带头结点的线性链表

2. 循环链表

循环链表是指链表的最后一个结点的指针值指向第一个结点，也就是说最后一个结点的指针值不再是空 Λ ，而是第一个结点的地址。如图 1.4 所示。

3. 可利用空间表

要在链表上执行插入和删除操作，就有一个新的结点从何处来又回到何处去的问题。为此，设立另一线性链表，称作可利用空间表。它与要操作的线性链表具有同样的

结构，所不同的是，它的结点信息域是空的。用一个指针 $FREE$ 指向可利用空间表，如图 1.5 所示。

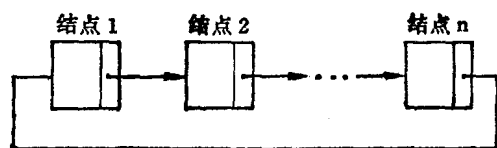


图 1.4 循环链表



图 1.5 可利用空间表

可利用空间表的作用是管理可用于链表插入的结点。当链表插入需要一个新结点时，就从可利用空间表中删除第一个结点，用这个结点去做链表插入；当从链表中删除一个结点时，就把这个结点插入到可利用空间表的第一个结点前面。

1.2.3 栈

栈是一种特殊的线性表。

栈是限定仅在表尾（表的一端）进行插入和删除运算的线性表，表尾称为栈顶（top），表头叫做栈底（bottom）。表中无元素时称为空栈。栈中有元素 $a_1, a_2, \dots, a_n$ ，如图 1.6 所示，称 a_1 是栈底元素。新元素进栈要置于 a_n 之上，删除或退栈必须先对 a_n 进行。这就形成了“后进先出”（LIFO）的操作原则。

栈的物理存储可以用顺序存储结构，也可用链式存储结构。

栈的运算除去插入和删除外，还有取栈顶元素，检查栈是否为空，清除（置空栈）等。

栈是使用最为广泛的数据结构之一，表达式求值、递归过程实现都是栈应用的典型例子。

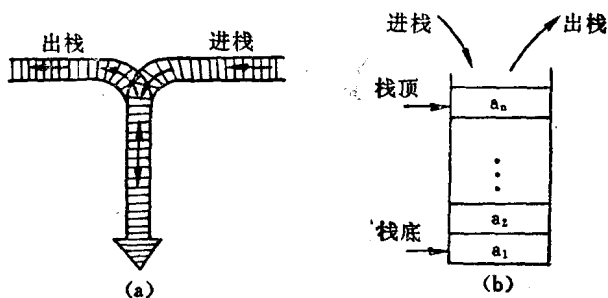


图 1.6 栈结构

1.2.4 队列

队列是一种特殊的线性表。

队列是限定所有的插入都在表的一端进行，所有的删除都在表的另一端进行的线性表。进行删除的一端叫队列的头，进行插入的一端叫队列的尾，如图 1.7 所示。在队列中，新元素总是加入到队尾，每次删除的总是队列头上的，即当前“最老的”元素，这就形成了先进先出（FIFO）的操作原则。

队列的物理存储可以用顺序存储结构，也可以用链式存储结构。

队列的运算除了插入和删除外，还有取队头元素，检查队列是否为空，清除（置空

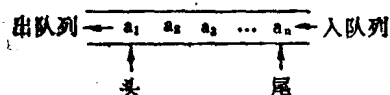


图 1.7 队列的示意

队列)等。

在顺序方式存储的队列中实现插入、删除运算时,若采取每插入一个元素,队尾指示变量 R 的值加 1,每删除一个元素,队头指示变量 F 的值加 1 的方法,则经过若干插入、删除运算后,尽管当前队列中的元素个数小于存储空间容量,但却可能无法再进行插入了,因为 R 已指向存储空间的末端。通常解决这个问题的方法是:把队列的存储空间从逻辑上看成一个环,当 R 指向存储空间的末端后,就把它重新置成指向存储空间的始端,如图 1.8 所示。

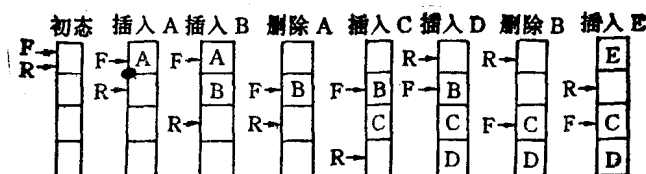


图 1.8 环状队列的插入和删除

队列在计算机中应用也十分广泛,硬设备中的各种排队器,缓冲区的循环使用技术,操作系统中的作业队列等都是队列应用的例子。

1.2.5 串

串(或字符串)是由零个或多个字符组成的有限序列,一般记为 $S = 'a_1a_2 \dots a_n'$,其中 S 是串的名字,用单引号括起来的字符序列是串的值。零个字符的串是空串。串中字符的数目就是串的长度。 a_i 是串中的字符,可以是字母、数字或其它字符。空串与空格构成的串如: ' ' 是不同的。

串的存储同样有顺序和链式两种。顺序存储时,又可以采用紧缩方式。图 1.9 为示意性说明。

串的基本运算有连接、赋值、求长度、全等比较、求子串、找子串位置以及替换等

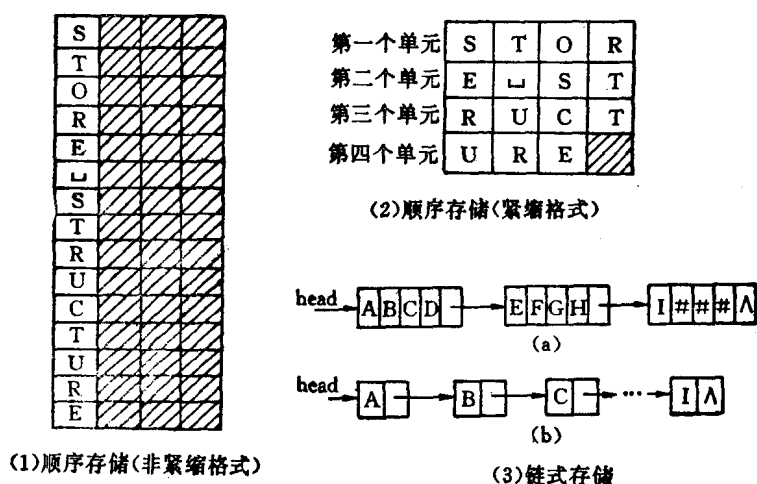
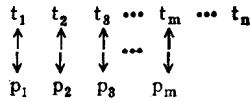


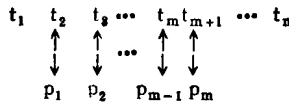
图 1.9 串的存储

7 种运算。所有这些运算的含义都是明显的。有些程序设计语言以标准子程序（函数或过程）方式提供了这些运算。

在串的基本运算中，找子串位置是非常重要的，因为在与正文处理有关的许多应用中都需要做模式匹配，即在一串正文 T 中找一个子串 P 的出现位置。模式匹配的最简单的做法是：用 P 中字符依次与 T 中字符比较，



如果 $t_1 = p_1, t_2 = p_2, \dots, t_m = p_m$ ，则匹配成功，找到了子串位置；否则将 P 右移一个字符，用 P 中字符从头开始与 T 中字符依次比较，



如此执行下去，直到某一步匹配成功，或者一直将 P 移到无法与 T 继续比较为止，则匹配失败。上述的匹配算法非常简单，但效率很低，在最坏的情况下，总的字符比较次数为 $m * (n - m + 1)$ 。因而在实际应用中，有多种改进的模式匹配算法。

1.3 树与二叉树

树形结构是一类重要的非线性结构，树和二叉树是最常用的树形结构。

1.3.1 树和二叉树的定义

树是一个或多个结点组成的有限集合 T ，有一个特定的结点称为根，其余的结点分为 $m(m \geq 0)$ 个不相交的集合 $T_1, T_2, \dots, T_m$ 。每个集合又是一棵树，被称作这个根的子树。

树形结构常用的术语有：

双亲、子女：结点的各子树的根称作该结点的子女；相应的结点称作其子女的双亲。

兄弟：具有相同双亲的结点互为兄弟。

结点的度：一个结点的子树的个数。

树叶：度为 0 的结点。

分支结点：度不为 0 的结点。

结点的层数：根结点的层数为 0，其他任何结点的层数等于它的双亲结点的层数加 1。

树林：0 棵或多棵不相交的树的集合（通常是有序集）。

图 1.10 所示的是一棵树。

二叉树是树形结构的另一个重要类型。

二叉树是结点的有限集合，这个有限集合或者为空集，或者由一个根结点及两棵不

相交的，分别称作这个根的左子树和右子树的二叉树组成。见图 1.11 所示。

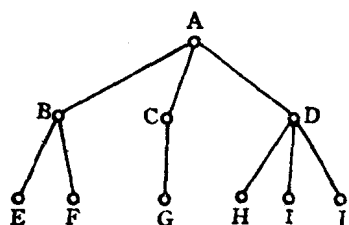


图 1.10 树

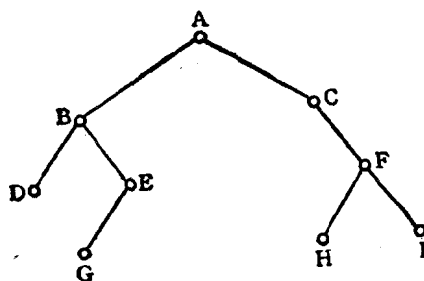


图 1.11 二叉树

特别要注意的是，二叉树不是树的特殊情况，尽管树和二叉树的概念之间有许多关系，但它们是两个概念。树和二叉树之间最主要的差别是：二叉树的结点的子树要区分为左子树和右子树，即使在结点只有一棵子树的情况下也要明确指出该子树是左子树还是右子树。

1.3.2 树的二叉树表示

在树（树林）与二叉树之间有一个自然的一一对应的关系，每一棵都能唯一地转换到它所对应的二叉树。

有一个自然的方式把树和树林转换成对应的二叉树：凡是兄弟就用线连起来，然后去掉双亲到子女的连线，只留下双亲到第一个子女的连线不去掉。对图 1.10 所示的树用上述方式处理后稍加倾斜，就得到对应的二叉树，如图 1.12 所示。

树所对应的二叉树里，一个结点的左子女是它在原来的树里的第一个子女，右子女是它在原来的树里的下一个兄弟。

树的二叉树表示对于树的存储和运算有很大意义，可以把对于树的许多处理转移到对应的二叉树中去做。

1.3.3 二叉树和树的周游（遍历）

周游（或称遍历）是树形结构的一种重要运算。周游一个树形结构就是按一定的次序系统地访问该结构中的所有结点，使每个结点恰被访问一次。可以按多种不同的次序周游树形结构。

1. 二叉树的周游

考虑到二叉树的基本组成部分是：根（N）、左子树（L）、右子树（R），因此可有 NLR、LNR、LRN、NRL、RNL、RLN 六种周游次序。通常使用前三种，即限定先左后右。这三种周游次序的递归定义如下。

- (1) 前序法（NLR 次序）：访问根，按前序周游左子树，按前序周游右子树。
- (2) 后序法（LRN 次序）：按后序周游左子树，按后序周游右子树，访问根。

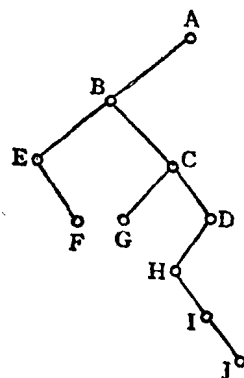


图 1.12 树对应的
二叉树