

中国计算机软件专业技术资格和水平考试

统编辅导教材

程序员级 高级程序员级 软件知识

第二版

施伯乐 主编



清华大学出版社

TP31
SBL/2

中国计算机软件专业技术资格和水平考试统编辅导教材

程序员级高级程序员级软件知识

(第二版)

施伯乐 主编

清华大学出版社

(京)新登 158 号

内 容 简 介

本书是中国计算机软件专业技术资格和水平考试委员会组织编写的考试统编辅导教材之一。这次再版以 1996 年 4 月颁布的《中国计算机软件专业技术资格和水平考试考试大纲》为依据,对程序员级和高级程序员级应必备的软件知识作了详尽的讲解。

主要内容包括:数据结构、操作系统、数据库系统、程序语言和语言处理程序、计算机网络、软件工程、计算机安全知识、软件开发工具平台和环境等计算机软件领域的基本概念、典型方法和技术。对考生应考有重要的参考价值。

本书可作为参加中国计算机软件专业技术资格和水平考试的考生备考的主要参考书,并可供各部门举办辅导班作为教材。也可作为大专院校师生和广大工程技术人员学习参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

程序员级高级程序员级软件知识/施伯乐主编. —2 版. —北京:清华大学出版社,1996

中国计算机软件专业技术资格和水平考试统编辅导教材

ISBN 7-302-02176-7/TP · 1038

I. 程… II. 施… III. 电子计算机-知识-教材 IV. TP303

中国版本图书馆 CIP 数据核字(96)第 07567 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:顺义振华印刷厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:23.25 字数:579 千字

版 次:1996 年 5 月第 2 版 1996 年 8 月第 3 次印刷

书 号:ISBN 7-302-02176-7/TP · 1038

印 数:30001—40000

定 价:29.50 元

出版说明

当今国际间的竞争是综合国力的竞争,关键又是科学技术的竞争,说到底人才的竞争,而电子信息技术又是当今国际间竞争的热点,它渗透和影响着现代化社会生活的各个方面,而科学技术(包括电子信息技术)是要靠人去掌握、去应用、去发展,国家的强盛、民族的振兴靠人才,人才的培养靠教育,所以,党中央要求我们把经济建设转到依靠科技进步和提高劳动者素质的轨道上来。

培养人才要坚持多种形式、多种途径,大力开展岗位培训,不断提高职工队伍的技术和专业水平,计算机软件专业技术资格和水平考试制度,就是为了加速我国电子信息技术的广泛应用和软件事业的发展,科学考核和合理使用人才,促进计算机软件人才的国际交流与合作,进一步深化职称改革。这种考试是于1985年首先在上海、云南、四川三省市实行的,1987年发展为部分省、区、市的联合考试,到1988年全国已有31个省、区、直辖市和计划单列市参加程序员、高级程序员两个级别的联合考试,1989年发展为程序员级、高级程序员级、系统分析员级三个级别的联合考试,1990年在全国统一组织实施了软件专业技术职务任职资格(水平)考试。1991年3月在计算机软件专业技术资格和水平考试工作会议上,对考试《暂行规定》作了修改,1991年的软件专业技术资格和水平考试就是依照新的《暂行规定》进行的。实践证明这是一种严格的认定考试制度,给应试者提供了一次均等的机会。软件专业技术资格和水平考试,每年举行一次,实行全国统一组织、统一大纲、统一试题、统一评分标准。

考试和培训是相辅相成的为了使全国参加统一考试的考生得到较好的辅导教材,编一套全国统一的辅导教材是完全必要的。1991年9月考委会在北京主持召开了辅导教材编审委员会会议,对统编辅导教材的编写、出版作了部署,成立了统编辅导教材编审委员会,编辑出版了《中国计算机软件专业技术资格和水平考试统编辅导教材》一套六册。1995年11月考试中心又在北京主持召开由教材编审委员会主编、副主编及各本教材主编、主审等有关专家会议。与会专家一致认为,1991年编辑出版的《中国计算机软件专业技术资格和水平考试统编辅导教材》,是一套深受读者欢迎的好教材,对培训计算机软件人才起到了良好的作用。根据4年来的培训实践及计算机技术的新发展并根据修订后的《考试大纲》对原书的内容进行部分补充、删改,并将原来一套六册,改成一套五册。他们是《初级程序员级软硬件知识》(主编:刘英 主审:吴克忠 沈林兴)、《程序员级高级程序员级软件知识》(主编:施伯乐 主审:吴立德)、《程序员级高级程序员级硬件知识》(主编:王爱英 主审:曲庭维)、《程序员级高级程序员级程序设计》(主编:张福炎 主审:郑国樑)、《1994年度—1995年度试题分析与解答(初级程序员级、程序员级、高级程序员级、系统分析员级)》(主编:王春森 主审:张然)。我们的本意是要把这套全国统编辅导教材编成为具有正确性、科学性、系统性,而且针对性强,在一段时间内相对稳定的好教材。由于编者水平所限及科技的快速发展,错误在所难免,请有关专家和读者给予指正。

教材编审委员会

主 编：陈正清

副主编：华平澜 陈祥禄 宋小海 罗晓沛 施伯乐 吴立德

编 委：(按姓氏笔划排列)

方 裕 王勇领 王春森 王爱英 王 珊 吕文超

冯大本 刘 英 刘福滋 朱慧真 吴克忠 郑人杰

周明德 张 然 张公忠 张吉锋 张福炎 侯炳辉

钱士钧 徐国平 徐国定 徐洁磐 唐 敏

秘书长：邵祖英

秘书组：王 永 邓小敏 尹芳平 赵永红 吕跃军

秘书组联络地点：北京海淀区学院南路 55 号电子部计算机技术培训中心
(邮编：100081)

前 言

《程序员级高级程序员级软件知识》一书涉及面很广,每一章的内容,在大学里往往是一学期的课程内容,因此编写的难度较高,为此在1991年底邀请了各方面的专家共同编写这本教材。这些专家都是长期活跃在教学第一线的有名望学者,编写的内容尽量做到既有科学性又能深入浅出,自1992年5月出版以来,深受计算机工作者的欢迎。

1995年底,考试大纲作了修改。为了使本教材继续符合大纲,并且能跟上计算机软件发展的形势,我们又邀请了原作者参与改写工作。第二版在内容上作了较大修改、充实和更新。下面就各章的内容作简要说明。

第1章数据结构是计算机软件学科的基础知识,由北京大学杨冬青撰写。本章介绍了线性表、集合、树、图和排序等内容,并对有关的重要算法作了阐述。对于算法,望读者结合有关的程序语言加以实现。

第2章操作系统是计算机中最基本的系统软件,由北京大学方裕撰写。本章介绍了对操作系统的核心及存储管理、文件管理、设备管理、作业管理等主要功能。读者通过学习,能全面理解操作系统的原理和内容。

第3章数据库管理在数据处理中有广泛的应用,由复旦大学丁宝康、施伯乐撰写。本章介绍了数据模型、数据库体系结构、关系代数操作、SQL语言及数据库设计。读者在建立数据库时应尽量应用数据库设计方法。

第4章程序语言和语言处理程序是理论性较强的一章,由华东师范大学徐国定撰写。本章介绍了程序语言的基础知识和汇编、编译、解释程序的基本原理,重点在编译方面。使读者对编译有一个全面的了解。

第5章计算机网络是“信息高速公路”的基础。第一版是北京工业大学刘福滋和清华大学张公忠撰写的,这次由复旦大学高传善进行了改写。本章介绍了开放系统互连的模型、局域网、公用数据网等基本知识,使读者对常用网络操作系统和新技术有一个详细的了解。

第6章软件工程是软件产业的重要支柱,由清华大学郑人杰撰写。本章以软件生存期为导引,介绍了软件开发计划、需求分析、设计、测试维护等方面的内容,使读者对软件开发的全过程有所了解,通过实践,进一步掌握有关的方法。

第7章计算机安全知识是软件知识的重要内容之一,由北京大学朱慧真撰写。本章介绍了计算机病毒、病毒的治理及预防知识,使读者对当前流行的计算机病毒有一个全面的了解。

第8章软件开发工具、平台和环境是新增加的一章,由复旦大学钱洛秋、刘畅撰写。本章介绍了软件开发活动中的常用开发工具、平台和环境。使读者对这些技术有一个全面的了解。

本书第二版仍由复旦大学施伯乐教授任主编,主审为复旦大学吴立德教授。

书中欠妥之处,欢迎专家和读者批评指正。

作 者

1996年1月31日

• II •

目 录

第1章 数据结构	1	1.7.1 顺序查找.....	28
1.1 概述.....	1	1.7.2 二分法查找.....	29
1.1.1 数据.....	1	1.7.3 分块查找.....	29
1.1.2 数据结构.....	1	1.7.4 散列表的存储和查找.....	29
1.1.3 主要的数据存储方式.....	2	1.7.5 二叉排序树.....	31
1.2 线性表.....	3	1.7.6 B-树和B ⁺ -树.....	33
1.2.1 线性表的基本运算.....	3	1.8 排序	34
1.2.2 顺序表和一维数组.....	3	1.8.1 插入排序.....	34
1.2.3 链表.....	3	1.8.2 选择排序.....	35
1.2.4 栈.....	9	1.8.3 交换排序.....	36
1.2.5 队列.....	9	1.8.4 基数排序.....	38
1.2.6 串.....	10	1.8.5 归并排序.....	38
1.3 多维数组、稀疏矩阵和广义表.....	11	第2章 操作系统	41
1.3.1 多维数组的顺序存储.....	11	2.1 概述.....	41
1.3.2 稀疏矩阵的存储.....	11	2.1.1 操作系统的工作原理.....	41
1.3.3 广义表的定义和存储.....	14	2.1.2 操作系统的硬件基础.....	43
1.4 集合.....	15	2.1.3 操作系统的分类.....	44
1.4.1 集合的概念.....	15	2.1.4 研究操作系统的观点.....	46
1.4.2 集合的运算.....	16	2.2 系统核心.....	48
1.4.3 集合的存储表示.....	17	2.2.1 进程.....	48
1.4.4 典型的集合结构.....	17	2.2.2 进程管理.....	55
1.5 树型结构.....	18	2.2.3 中断接收.....	57
1.5.1 树的定义.....	18	2.2.4 时钟管理.....	57
1.5.2 二叉树的定义.....	18	2.3 存储管理.....	58
1.5.3 树的二叉树表示.....	19	2.3.1 地址映射.....	58
1.5.4 二叉树和树的周游.....	19	2.3.2 段式管理.....	59
1.5.5 二叉树的存储和线索 二叉树.....	20	2.3.3 页式管理.....	60
1.5.6 二叉树周游算法.....	22	2.3.4 段页式管理.....	60
1.6 图.....	23	2.3.5 虚存技术.....	61
1.6.1 图的概念及术语.....	23	2.3.6 Cache 管理.....	65
1.6.2 图的存储.....	24	2.4 文件管理.....	66
1.6.3 图的周游和生成树.....	25	2.4.1 概述.....	66
1.6.4 最短路径.....	27	2.4.2 文件控制块与文件目录.....	67
1.6.5 拓扑排序.....	27	2.4.3 文件的物理组织.....	68
1.6.6 关键路径.....	28	2.4.4 文件系统的使用.....	70
1.7 查找.....	28	2.4.5 外存管理.....	72
		2.5 设备管理.....	75

2.5.1 设备管理与文件系统	75	3.6.5 SQL 的访问控制	123
2.5.2 设备连接	76	3.6.6 嵌入式 SQL	123
2.5.3 外部设备的种类	76	3.7 数据库设计	125
2.5.4 虚设备技术	77	3.7.1 数据库设计的全过程	125
2.5.5 输入/输出转向	77	3.7.2 关系数据库的模式设计 理论	128
2.6 作业管理	78	3.8 数据库的保护	131
2.6.1 作业处理方式	78	3.8.1 数据库的恢复	131
2.6.2 批处理方式下的作业管理	79	3.8.2 数据库的完整性	132
2.7 其它管理	82	3.8.3 数据库的并发控制	134
2.7.1 死锁问题	82	3.8.4 数据库的安全性	137
2.7.2 保护与安全	84	3.9 常用的 DBMS	138
2.7.3 操作系统的结构	85	3.9.1 微型型 DBMS	138
2.7.4 性能分析	86	3.9.2 主流型 DBMS	140
2.8 实例分析	86	第4章 程序语言和语言处理程序	144
2.8.1 UNIX 操作系统	86	4.1 引言	144
2.8.2 Windows NT	89	4.2 程序语言基础知识	145
2.8.3 DOS	92	4.2.1 概述	145
第3章 数据库系统	94	4.2.2 语言所提供的数据结构	148
3.1 数据管理技术的发展	94	4.2.3 语言所提供的控制结构	154
3.1.1 数据管理技术的发展阶段	94	4.2.4 语言所提供的模块结构	162
3.1.2 数据库技术的基本概念	95	4.3 汇编程序基本原理	164
3.1.3 数据库系统的进一步发展	96	4.3.1 机器语言、汇编语言	164
3.2 数据和联系的描述	96	4.3.2 汇编程序	166
3.2.1 数据的描述	96	4.3.3 装配程序	168
3.2.2 数据联系的描述	97	4.3.4 宏指令	170
3.3 数据模型	98	4.4 编译程序基本原理	170
3.3.1 数据模型的定义	98	4.4.1 编译程序概貌	170
3.3.2 概念数据模型	99	4.4.2 形式语言基础	172
3.3.3 基本数据模型	100	4.4.3 语法分析	178
3.4 数据库的体系结构	103	4.4.4 句法分析	181
3.4.1 数据库的数据体系结构	103	4.4.5 句法制导翻译	189
3.4.2 数据库系统	105	4.4.6 代码生成	192
3.4.3 数据库管理系统	106	4.4.7 代码优化	196
3.4.4 用户访问数据的过程	108	4.5 解释程序基本原理	198
3.5 关系模型和关系运算	109	第5章 计算机网络	200
3.5.1 关系模型的基本概念	109	5.1 计算机网络的演变和发展	200
3.5.2 关系运算	110	5.2 开放系统互连基本参考模型	206
3.5.3 关系数据库 DML 的特点	115	5.2.1 层次模型	206
3.6 关系数据库 SQL 语言	115	5.2.2 各层的主要功能	206
3.6.1 SQL 数据库的体系结构	115	5.2.3 ISO 与 ITU 有关的国际 标准	209
3.6.2 SQL DDL	116	5.3 局部区域网	210
3.6.3 SQL DML 的数据查询	117		
3.6.4 SQL DML 的数据更新	122		

5.3.1	局域网的主要特点及分类	210	6.5.1	结构化程序	275
5.3.2	IEEE-802 和 ISO 8802 标准	211	6.5.2	程序设计风格	276
5.3.3	CSMA/CD 总线网	213	6.6	软件测试	279
5.3.4	令牌总线网	215	6.6.1	软件测试的基本概述	279
5.3.5	令牌环网	217	6.6.2	测试步骤	280
5.3.6	令牌环网	218	6.6.3	测试用例设计方法简介	281
5.4	公用数据网与 CHINAPAC	218	6.7	软件维护	288
5.4.1	X.3-X.25 等有关标准	218	6.7.1	什么是软件维护	289
5.4.2	中国公用数据网 CHINAPAC	219	6.7.2	维护工作存在的问题及其分析	291
5.5	常用网络操作系统	221	6.7.3	可维护性	292
5.5.1	NetWare 与 LAN Manager	221	6.7.4	软件维护的管理	293
5.5.2	NetWare 的安装	223	6.8	软件工程的文档编制	295
5.5.3	NetWare 实用命令	226	6.8.1	软件文档的作用和分类	295
5.5.4	基于 TCO/IP 的网络软件	228	6.8.2	文档编制的质量要求	298
5.6	高速网络新技术	230	6.8.3	文档的管理和维护	299
5.6.1	光纤分布数据接口 FDDI	230	6.9	软件工程标准化	299
5.6.2	快速以太网	235	6.9.1	什么是软件工程标准化	300
5.6.3	帧中继	238	6.9.2	软件工程标准化给软件工作带来的好处	301
5.6.4	异步传输模式 ATM	242	6.9.3	软件标准的制定与推行	301
5.6.5	分布式队列双总线子网 DQDB	246	6.9.4	软件工程标准的层次	302
第 6 章	软件工程	252	6.10	软件管理	302
6.1	软件工程概述	252	6.10.1	软件项目的特点与软件管理职能	303
6.1.1	软件技术发展中的难题——软件危机	252	6.10.2	制定计划	304
6.1.2	软件研制工程化	253	6.10.3	软件人员的组织管理	306
6.1.3	软件的生存期	253	6.10.4	指导与检验	307
6.2	软件开发计划的制定	255	第 7 章	计算机安全知识	309
6.2.1	系统定义及分析	255	7.1	计算机安全知识概述	309
6.2.2	可行性研究	255	7.1.1	计算机安全的定义	309
6.2.3	实施计划报告的编写	256	7.1.2	计算机危害	309
6.3	软件需求分析	256	7.1.3	计算机安全治理的范围	311
6.3.1	需求分析阶段的任务	257	7.2	计算机病毒的基础知识	312
6.3.2	结构化分析方法	257	7.2.1	计算机病毒的定义	313
6.4	软件设计	263	7.2.2	计算机病毒的起源	313
6.4.1	结构化设计方法	263	7.2.3	计算机病毒的特征	314
6.4.2	Parnas 方法概述	266	7.2.4	计算机病毒的种类	315
6.4.3	Jackson 方法概述	268	7.3	计算机病毒的作用机理	316
6.4.4	详细设计中的算法描述	269	7.3.1	计算机病毒的简单结构	316
6.5	结构化程序设计与程序设计风格	274	7.3.2	计算机病毒的寄生方式	317
			7.3.3	计算机病毒的传染机制	318
			7.3.4	计算机病毒的破坏作用	321

7.3.5 计算机病毒举例	323	8.2.3 需求分析工具	344
7.4 计算机病毒的治理	324	8.2.4 设计工具	346
7.4.1 计算机病毒的检测	324	8.2.5 编码和排错工具	347
7.4.2 计算机病毒的消除	329	8.2.6 测试工具	349
7.4.3 计算机病毒的预防	329	8.2.7 维护工具	350
7.4.4 PC-DOS 3.3 的参照信息 ...	331	8.2.8 数据库应用开发工具	351
7.5 计算机病毒特征简介	335	8.2.9 用户界面开发工具	352
7.6 我国计算机信息系统安全条规	339	8.2.10 软件开发工具的评价 和选择	352
第8章 软件开发工具、平台和环境	342	8.3 软件开发平台和环境	353
8.1 引言	342	8.3.1 通用软件开发平台	353
8.2 软件开发工具	343	8.3.2 专用软件开发平台	354
8.2.1 概述	343	8.3.3 软件开发环境	357
8.2.2 项目管理工具和软件配置 管理工具	344	附录 软件系统的新发展	360

第1章 数据结构

“数据结构”是计算机各有关专业的一门重要基础课程。计算机科学领域中,尤其是系统软件和应用软件的设计和实现中都要用到各种数据结构。学习“数据结构”既为进一步学习其他软件课程提供必要的准备知识,又有助于提高软件设计和程序编制水平。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,以及在数据结构上进行各种运算的执行程序,并作简单的算法分析。

1.1 概 述

1.1.1 数据

数据就是对现实世界的事物采用计算机能够识别、存储和处理的方式(例如数字、字符等)进行的描述。简言之,数据就是计算机化的信息。

数据元素是数据的基本单位,即数据集中的个体。有些情况下也把数据元素称作结点、记录、表目等。一个数据元素可由一个或多个数据项组成,数据项是有独立含义的数据最小单位。有时也把数据项称作域、字段等。例如,可以将一个学生的有关信息作为一个数据元素,它由姓名、专业、学号等数据项组成。

1.1.2 数据结构

被计算机加工的数据元素不是互相孤立的,它们彼此间一般存在着某些逻辑上的联系,这些联系需要在对数据进行存储和加工时反映出来。因此,数据结构概念一般包括三个方面的内容:数据之间的逻辑关系、数据在计算机中的存储方式、以及在这些数据上定义的运算的集合。

1. 数据的逻辑结构

数据的逻辑结构只抽象地反映数据元素间的逻辑关系,而不管其在计算机中的存储表示方式。

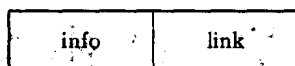
数据的逻辑结构分为线性结构和非线性结构。若各数据元素之间的逻辑关系可以用一个线性序列简单地表示出来,则称之为线性结构,否则称为非线性结构。线性表是典型的线性结构,而树、图等都是非线性结构。

2. 数据的存储结构

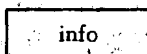
数据的存储结构是逻辑结构在计算机存储器里的实现。

为全面地表示一个逻辑结构,它在存储器中的映象应包括数据元素自身值的表示和数据元素之间的关系的表示两个方面。因此,存储在计算机中的数据结构,其结点的各域按性质可分成两大类,一类是存放自身值的域,例如姓名、专业、学号等,通常称之为自身信息域,可用标识符 info 表示这些域的全体;另一类是存放该结点与其它结点的关系的域,例如一

个或多个指针,或其它形式的连接信息,通常称之为连接信息域,可用标识符 link 表示这些域的全体。一般情况下,存储结构中结点的形式为:



这里说的是一般情况,而在某些存储结构中,结点可以不包括连接信息域,这时的结点形式为:



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。

1.1.3 主要的数据存储方式

有多种不同的方式来实现数据的逻辑结构到计算机存储器的映象。下面介绍两类最主要的存储方式,大多数数据结构的存储表示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实现。

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。

除上述两种主要存储方式外,散列法也是在线性表和集合的存储表示中常用的一种重要存储方式,我们将在 1.7.4 节中介绍。

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。若线性表中的元素都是单个字符,则称作串。

1.2.1 线性表的基本运算

线性表常用的运算分成四类,每类包含若干种运算。

1. 查找

- ① 查找线性表中第 i 个结点的值。
- ② 在线性表中查找值为 x 的结点。

2. 插入

- ① 把新结点插在线性表的第 i 个结点位置上。
- ② 把新结点插在值为 x 的结点的前面(或后面)。

3. 删除

- ① 在线性表中删除第 i 个结点。
- ② 在线性表中删除值为 x 的结点。

4. 其它运算

- ① 统计线性表中结点的个数。
- ② 遍历线性表中所有结点。
- ③ 把一个线性表拆成几个线性表。
- ④ 把几个线性表合并成一个线性表。
- ⑤ 根据结点的某个字段值升序(或降序)重新排列线性表。

1.2.2 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,可能需要往后移动一系列结点。若顺序表中结点个数为 n ,且往每个位置插入的概率相等,则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 $n/2$ 。

1.2.3 链表

1. 线性链表(单链表)

线性链表就是链式存储的线性表,它的每个结点中含有一个指针域,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。

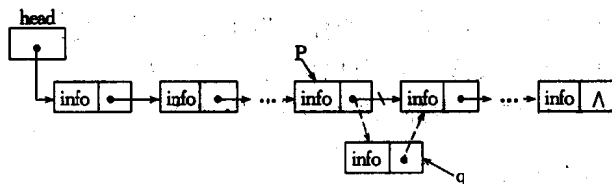


图 1.1 单链表的插入

插入运算的关键步骤为:

$q \uparrow .link := P \uparrow .link;$

$P \uparrow .link := q;$

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。

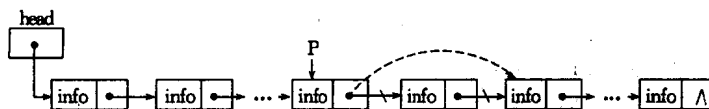


图 1.2 单链表的删除

删除运算的关键步骤为:

$q := P \uparrow .link;$

$P \uparrow .link := q \uparrow .link;$

注意,做删除运算时改变的是被删结点的前一个结点中指针域的值。因此,若要求查找且删除某一结点,则应在查找被删结点的同时,记下它的前一个结点的位置。

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,而只改变头结点的指针域的值,因此就可以和其它位置的插入、删除同样处理了。

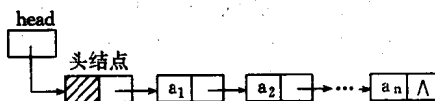


图 1.3 带头结点的线性链表

如果让线性链表的最后一个结点的指针指向第一个结点,便可得到一个环形链表。图 1.4 给出了带头结点的环形链表的结构形式。在环形链表中,可以从其中某个结点出发访问到表中所有结点。

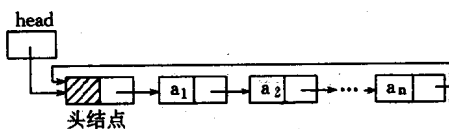


图 1.4 带头结点的环形链表

2. 双链表

在单链表中,从任何一个结点能通过 link 域找到它的后继,但不能找出它的前驱。如果在链表的每个结点中包括两个指针域,其中 rlink 指向结点的后继,llink 指向结点的前驱,就可以方便地进行向后和向前两个方向的查找了。这样的链表称作双链表,如图 1.5 所示。

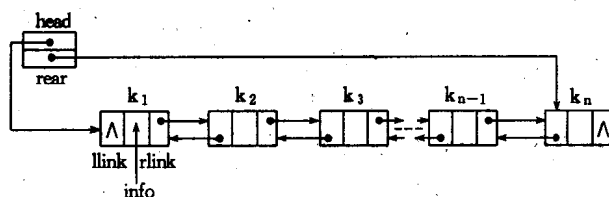


图 1.5 双链表

在双链表中,如果要删除指针变量所指的结点,只需修改该结点前驱的 rlink 字段和后继的 link 字段,即

$p \uparrow .llink \uparrow .rlink := p \uparrow .rlink;$

$p \uparrow .rlink \uparrow .llink := p \uparrow .llink;$

如果要在 P 所指结点后插入 q 所指的新结点,只需修改 P 所指结点的 rlink 字段和原后继的 llink 字段,并置 q 所指结点的 llink 和 rlink 值,即

$q \uparrow .llink := p;$

$q \uparrow .rlink := p \uparrow .rlink;$

$p \uparrow .rlink \uparrow .llink := q;$

$p \uparrow .rlink := q;$

插入和删除运算前后指针的变化情况如图 1.6 和如图 1.7 所示。

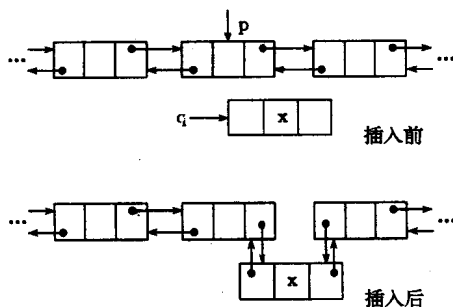


图 1.6 双链表的插入

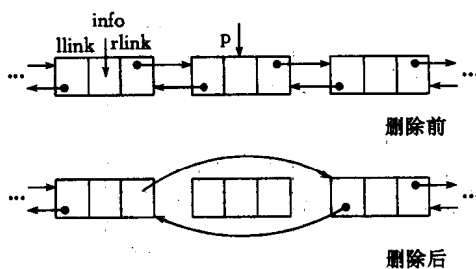


图 1.7 双链表的删除

和线性链表一样,双链表也有变形。图 1.8 给出环形的双向链表的结构形式。

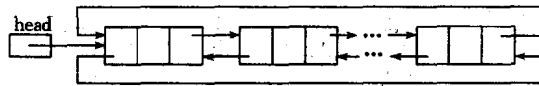


图 1.8 环形双向链表

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,设立另一种线性链表,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 1.9 所示。

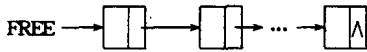


图 1.9 可利用空间表

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。

4. 两个例子

请仔细研究下面例子,它包括了链表的检索、插入、删除运算和可利用空间表的管理。

[例 1.1] 阅读下列对线性表进行操作的 3 个子程序的流程图,从供选择的答案中选出应该填入 a~e 处的字句。

在主存储器中有一个如图 1.10 所示的表格结构,表格的每个元素由值(v)和指针(p)两部分组成。在表格中以链接方式存放着一个线性表 L,它的第一个元素的位置存放在 LP 中。表格中全部空闲元素也链接成一个线性表 E,它的第一个元素的位置存放在 CP 中。两个线性表的最后一个元素的指针都是 0。

(1) 子程序 enter(u,m): 在线性表 L 中位于 m(m≠0)处的元素后插入一个以 u 为值的结点,如图 1.11 所示。

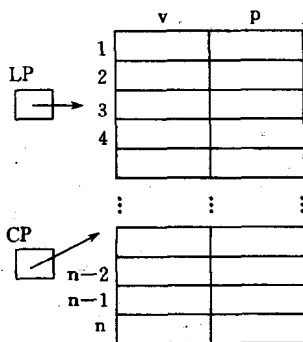


图 1.10 线性表 L 和线性表 E

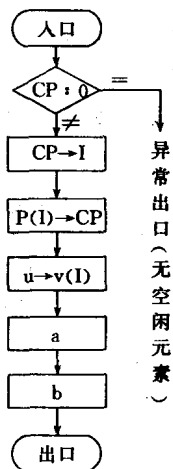


图 1.11 子程序 Enter

(2) 子程序 find(u, m): 从线性表 L 中找出第一个其值为 u 的元素, 把位置送入 m 。没有以 u 为值的元素时, 把 m 置为 0, 如图 1.12 所示。

(3) 子程序 delete(m): 从至少有两个元素的线性表 L 中删除紧接在位置 m 之后的一个元素, 把删除的元素放进线性表 E 中, 如图 1.13 所示。

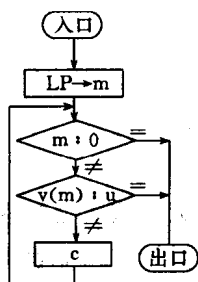


图 1.12 子程序 find

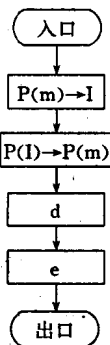


图 1.13 子程序 delete

供选择的答案

a, b, c, d, e; $m \rightarrow P(m)$	$P(m) \rightarrow m$	$P(m) \rightarrow P(I)$
$P(I) \rightarrow P(m)$	$CP \rightarrow P(I)$	$P(I) \rightarrow CP$
$I \rightarrow CP$	$CP \rightarrow I$	$I \rightarrow P(m)$
$P(m) \rightarrow I$		

〔分析〕 插入时所用的存储单元取自可利用空间表 E , 删除时释放的存储单元归还给可利用空间表 E 。

子程序 Enter 对链表 L 执行插入运算。新结点所要占用的存储单元从可利用空间表 E 中已取得, 其地址存放在 I 中, 并且新结点的信息值 u 也已置好。于是可知余下的工作是将此新结点插入到链表 L 中位于 m 处的结点之后, 即使新结点的指针指向 m 处结点的原后继结点, 而使 m 处结点的指针指向新结点。所以答案为:

a: $P(m) \rightarrow P(I)$ b: $I \rightarrow P(m)$

读者也可不难分析出其它答案:

c: $P(m) \rightarrow m$ d: $CP \rightarrow P(I)$ e: $I \rightarrow CP$

注意, 答案中 a 和 b 的次序, d 和 e 的次序都不能交换, 即有用的指针值在被使用之前不允许被破坏。

〔例 1.2〕 阅读下面关于约瑟夫 (Joseph) 问题的说明和流程图, 如图 1.14 所示, 在①~⑤处填上适当的操作语句。

〔说明〕 有 1 至 N 编号的 N 个人 ($N > 1$) 按顺时针方向围坐一圈, 每人持有一个密码 (正整数), 一开始以正整数 M 作为报数上限值。从第一个开始顺时针方向自 1 开始顺序报数, 报到 M 时停止报数, 报 M 的人出列, 将他的密码作为新的报数上限值。从他的顺时针方向上的下一个人开始重新报数, 如此下去, 直至所有的人全部出列为止。要求产生表示出列顺序的表。例如, 若 $N=7$, 每个人的密码依次是 3, 1, 7, 2, 4, 8, 4, M 的值为 20, 则出列顺序为 6, 1, 4, 7, 2, 3, 5。