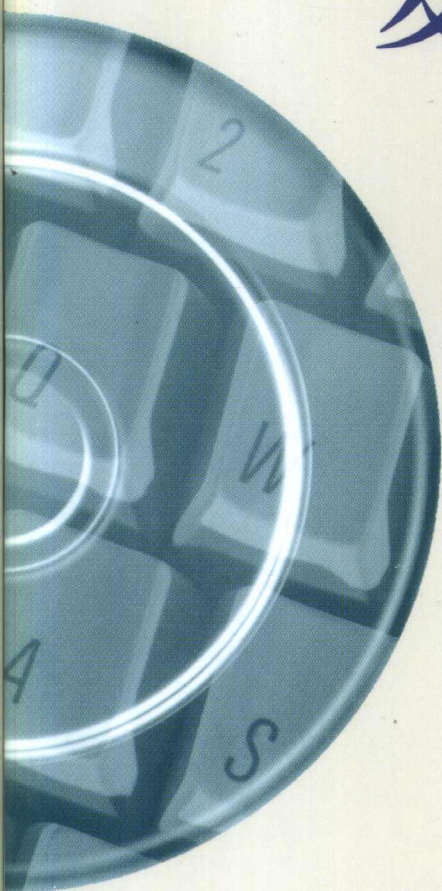


● 全国高等教育自学考试系列辅导教材

数据结构导论 复习与考试指导

计算机及应用专业（专科）

胡学钢 主编



高等教育出版社

913

TP311.12
H5361

全国高等教育自学考试系列辅导教材

数据结构导论 复习与考试指导

计算机及应用专业(专科)

胡学钢 主编



高等教育出版社

内容提要

本书是全国高等教育自学考试计算机及应用专业系列辅导教材之一,是与专科《数据结构导论》(陈小平主编)教材配套的辅导教材。本书按原教材的章节对各部分内容展开辅导,内容包括线性表、串、栈、队列、数组、树和二叉树、图、查找、文件和排序等。各章中除了采用通俗易懂的语言介绍有关概念、方法外,还通过大量例题及其分析展示课程内容的应用及其学习方法,因而容易激发学生的学习兴趣。书中包含许多思考题,并在附录中给出了涉及面广泛的模拟试卷以供学生自测,并提供了详细的解析。

本辅导教材除适用于计算机及应用专业考生以外,还适用于计算机信息管理、计算机通信工程和计算机应用及教育等专业相同课程的考生。

虽然本书是作为自学考试辅导教材而编写的,但由于其所具有的系统性和通俗性,因而也可作为高职高专层次数据结构课程的辅导教材。

图书在版编目(CIP)数据

数据结构导论复习与考试指导/胡学钢主编. —北京:

高等教育出版社, 2003. 2

ISBN 7-04-011560-3

I. 数... II. 胡... III. 数据结构-高等教育-自学考试-自学参考资料 IV. TP311.12

中国版本图书馆 CIP 数据核字(2003)第 000425 号

出版发行 高等教育出版社
社 址 北京市东城区沙滩后街 55 号
邮政编码 100009
传 真 010-64014048

购书热线 010-64054588
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

经 销 新华书店北京发行所
印 刷 中国农业出版社印刷厂

开 本 787×1092 1/16
印 张 12.75
字 数 290 000

版 次 2003 年 2 月第 1 版
印 次 2003 年 2 月第 1 次印刷
定 价 18.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

序

高等教育自学考试正式实施已经 20 周年了。20 年来,每年都有数以百万计的人参加考试,目前全国自考在籍考生已达 1600 多万人,显示了其在高等教育总体格局中的地位。近年来,随着计算机技术及应用水平的不断提高,报考“计算机及应用”专业的考生逐年增加。

为了满足自学考生的需求,高等教育出版社根据全国高等教育自学考试指导委员会 1999 年重新修订的“计算机及应用专业教学大纲”的教学基本要求和相应新教材的出版,组织编写了“高等教育自学考试计算机及应用专业系列辅导教材”。在这一系列辅导教材中,包括“计算机及应用”专业专科和独立本科段开考的共 15 门课程的辅导教材。

这套辅导教材是根据教材的编写思想和教学大纲的基本要求,从自考生的学习水平及自考生以自学为主的特点出发,为应试考生提供既包含自考教材中的考核知识点,又包含各知识点相互关系的讲解;既指出课程中的学习重点,又进行课程难点的分析;既提供多种典型例题,又详细地进行分析解答;各章后面附有大量的自测试题,以帮助考生测试各章内容的掌握情况,全书后面包含有多套模拟试卷,综合、全面测试考生对本课程的理解和掌握情况;使考生既学到了知识,又逐渐进入考试的实战状态,达到考试中应答自如、顺利通过考试的目的。本系列辅导教材并不单纯以解答教材中的例题为目的,而是通过作者在自考辅导过程中积累的典型例题,逐层分析、解答,从而使考生全面掌握本课程的知识。由此,也构成了本系列辅导教材例题丰富、实用性强的特点。

在组织编写这套系列辅导教材时,高等教育出版社选择的作者中,有些是自学考试教材的编写者,有些是多年来一直从事自学考试辅导教学工作的出色教师,他们非常了解自学考生的特点,富有自学考试的辅导经验。

本系列辅导教材适用于“计算机及应用”专业专科和独立本科段考生的自学及教师上课辅导,也适用于“计算机信息管理”专业、“计算机通信工程”专业、“计算机应用及教育”专业相同课程的考生,也可以作为工程技术人员、社会读者的学习用书。

全国高等教育自学考试指导委员会
电子电工与信息类专业委员会副主任
陈国良

2001 年 7 月

前 言

面向全国开考的计算机及应用专业(专科)自学考试,为广大考生提供了一个学习和掌握计算机科学与技术的重要途径。然而,要想实现既定的目标,通过自学完成各门课程的学习是需要付出艰辛劳动的,因为很少有课程能轻易掌握。

在计算机及应用专业(专科)的课程中,数据结构导论是重要的专业基础课程,是提高软件设计水平以及学习后续课程所必需的基础。课程中介绍了软件设计中常见的线性表、串、栈和队列、数组、树和二叉树、图、文件等数据结构及其在计算机内存中的存储结构和各种操作的实现,介绍了软件设计中常用的排序和查找方法,并讨论有关运算的性能。通过对这些内容的学习,将使学能熟练地掌握各种常用结构的特性,各种运算的实现方法及其性能,并能在实际应用中根据具体问题的要求设计出合理的数据结构和算法。

然而,由于该课程中内容多,而且许多内容较抽象,特别是其中大量的算法以及所用到的递归技术,一些深层次的技术等,同时由于缺乏有效的实验条件,使学习数据结构课程的难度较大。对于以自学为主的考生来说,学习数据结构课程更是困难重重。虽然教材较多,但由于受篇幅所限,学习效果不理想。

本书是针对全国高等教育自学考试计算机及应用专业(专科)数据结构导论(陈小平主编)编写的配套辅导教材,按原教材的章节顺序对各部分内容展开辅导。各章中除了采用通俗易懂的语言系统地介绍有关概念、方法外,还通过大量例题及其分析展示了课程内容的应用及其学习方法,因而容易激发学生的学习兴趣,收到良好的学习效果。这是本书的特色之一。最后部分给出了涉及面广泛的模拟考试试卷以供学生自测,并提供了详细的分析和讲解。

各部分的主要内容如下:

第一章介绍了数据结构课程的内容、特点以及一些所需的基础知识和学习方法,有助于整个课程的学习。

第二章介绍了线性表和串的逻辑结构和运算,重点讨论了顺序表和链表结构及其基本运算实现,并给出了大量例题及分析讲解。

第三章介绍了栈、队列和数组这三种结构的逻辑结构和运算,顺序栈、链栈、顺序(循环)队列和链队列及其基本运算实现,介绍了数组的有关内容。

第四章介绍了树的逻辑结构、存储结构和运算,介绍了二叉树的概念、性质、存储结构,重点讨论了二叉树的遍历运算方法及其应用,并给出了相应的例题及分析,另外还介绍了哈夫曼树的有关知识及其应用。

第五章介绍了图的有关概念,介绍了两种常用的存储结构,重点讨论了两种遍历算法的方法和应用,最后以通俗的方式介绍了图的几个应用算法。

第六章所介绍的查找是软件设计中常用的基本运算,本章讨论了在顺序表、树表和散列表等几种表结构上的查找方法及其性能。

第七章简要介绍文件的基本知识。

第八章重点讨论了各种排序的方法、算法及其性能分析。

书后附录中包含了三套模拟试卷。每套模拟考试试卷都约 40 题,涉及面大,是考生自测及复习的重要内容,其后的模拟试卷解析更详细、深入地介绍了所学知识及其应用的方法。

全书由几位作者合作完成,人员安排情况如下:

合肥工业大学计算机与信息学院胡学钢负责整书的策划、组织,并具体编写第一章、第二章、第四章、第五章、第七章和模拟试卷,并做了校阅全书,张晶编写第三章,阙夏编写第六章,周红鹃编写第八章。

虽然本书是作为计算机专科自学考试的辅导教材编写的,但由于所具有的系统性和通俗性,因而也可以作为高职高专层次数据结构课程的辅导教材,也有助于本科层次学生的学习。

由于受水平及时间所限,书中难免有错误和不足之处,敬请批评指导。

胡学钢

2002 年 12 月于合肥工大

目 录

第一章 概论	(1)	3.2.1 定义和运算	(47)
1.1 本章内容概述及学习方法	(1)	3.2.2 顺序队列	(48)
1.2 数据结构课程的作用、特点及学习方法	(1)	3.2.3 链队列	(51)
1.2.1 数据结构课程在计算机专业中的作用	(1)	3.3 数组	(53)
1.2.2 数据结构课程的主要内容和特点	(3)	3.3.1 数组的定义和运算	(53)
1.2.3 数据结构课程的学习方法	(3)	3.3.2 数组顺序存储	(54)
1.3 本章内容学习和辅导	(4)	3.3.3 矩阵的压缩存储	(55)
1.3.1 有关概念的理解	(4)	第四章 树	(57)
1.3.2 算法描述及分析	(6)	4.1 本章内容概述	(57)
1.4 复习题	(8)	4.1.1 本章知识点概述	(57)
第二章 线性表	(9)	4.1.2 树结构的有关概念	(58)
2.1 本章内容概述及学习方法	(9)	4.1.3 二叉树的有关概念、性质及存储结构	(59)
2.1.1 内容简介	(9)	4.2 重点、难点分析	(64)
2.1.2 学习方法建议	(10)	4.2.1 二叉树的遍历	(64)
2.2 线性表的定义和运算	(10)	4.2.2 树和森林	(76)
2.2.1 线性表的定义	(10)	4.2.3 哈夫曼树	(83)
2.2.2 线性表的运算	(11)	第五章 图	(89)
2.3 线性表的顺序表存储结构	(12)	5.1 本章内容概述及学习方法	(89)
2.3.1 顺序存储结构	(12)	5.1.1 引言	(89)
2.3.2 顺序表运算讨论的实现	(13)	5.1.2 本章学习方法	(89)
2.3.3 顺序表的应用	(16)	5.2 基础知识	(90)
2.4 链表	(24)	5.2.1 基本概念	(90)
2.4.1 链表结构	(24)	5.2.2 图的存储结构	(92)
2.4.2 链表运算实现	(27)	5.3 图的遍历算法及其应用	(95)
2.4.3 其他形式的链表结构	(40)	5.3.1 深度优先搜索遍历算法及其应用	(95)
2.5 串	(41)	5.3.2 广度优先搜索遍历算法及其应用	(101)
2.5.1 串的定义和运算	(41)	5.4 图的应用	(105)
2.5.2 串的存储	(42)	5.4.1 最小生成树	(106)
第三章 栈、队列和数组	(44)	5.4.2 最短路径	(110)
3.1 栈	(44)	5.4.3 拓扑排序	(114)
3.1.1 定义和运算	(44)	第六章 查找表	(117)
3.1.2 顺序栈	(45)	6.1 本章内容概述及学习方法	(117)
3.1.3 链栈	(47)	6.1.1 概述	(117)
3.2 队列	(47)	6.1.2 基本概念	(117)

6.1.3 学习方法	(118)	8.1 本章内容概述及学习方法	(135)
6.2 顺序表的查找	(118)	8.1.1 概述	(135)
6.2.1 简单顺序查找	(118)	8.1.2 排序及其分类	(135)
6.2.2 有序表的二分查找	(119)	8.1.3 排序算法分析指标	(136)
6.2.3 索引顺序表的查找	(122)	8.1.4 本章的学习方法	(136)
6.3 二叉排序树的查找	(123)	8.2 直接插入排序	(136)
6.3.1 二叉排序树及其查找	(123)	8.3 交换排序	(138)
6.3.2 二叉排序树的构造和插入	(124)	8.3.1 冒泡排序	(138)
6.4 散列表的查找	(126)	8.3.2 快速排序	(140)
6.4.1 哈希表的有关概念	(126)	8.4 选择排序	(144)
6.4.2 哈希函数的构造方法	(126)	8.4.1 直接选择排序	(144)
6.4.3 处理冲突的方法	(127)	8.4.2 堆排序	(145)
6.4.4 散列表的查找	(129)	8.5 归并排序	(150)
第七章 文件	(131)	8.5.1 归并	(150)
7.1 概述	(131)	8.5.2 归并排序	(151)
7.1.1 文件的逻辑结构	(131)	附录一 模拟试卷	(152)
7.1.2 文件的运算	(131)	模拟试卷一	(152)
7.1.3 文件的存储介质	(132)	模拟试卷二	(155)
7.2 常见文件组织形式	(132)	模拟试卷三	(158)
7.2.1 顺序文件	(133)	模拟试卷四	(162)
7.2.2 索引文件	(133)	附录二 模拟试卷解析	(165)
7.2.3 ISAM 文件	(133)	模拟试卷一解析	(165)
7.2.4 VSAM 文件	(133)	模拟试卷二解析	(173)
7.2.5 散列文件	(134)	模拟试卷三解析	(180)
7.2.6 多关键字文件	(134)	模拟试卷四解析	(187)
7.3 本章思考题	(134)	参考文献	(192)
第八章 排序	(135)		

第一章 概 论

1.1 本章内容概述及学习方法

本章主要介绍与整个数据结构课程有关的概念、术语、算法及其描述语言和分析等。尽管考试大纲只要求“了解”概念,但为了后面内容学习的需要,还是希望考生能深入理解这些概念。虽然本章是入门性的内容,但由于概念较多,并且相对较抽象,因此,还是有一定学习难度的。

考虑到本章内容及整个课程的特点,在本章的辅导中将介绍以下一些内容:

(1) 数据结构课程的作用、特点及学习方法,如果能了解这些特点和作用,将对整个课程的学习有很大帮助。

(2) 本章有关的概念和学习方法。

(3) 算法描述语言和分析。

通过对这些内容的学习,希望使初学者不仅能理解数据结构课程的作用和特点,同时还能理解有关的概念和术语,掌握算法的描述和分析的方法,为后续课程的学习打好基础。

1.2 数据结构课程的作用、特点及学习方法

1.2.1 数据结构课程在计算机专业中的作用

许多初学者在学习一门课程时,大多会提出这样一些问题:学这门课程有什么用?如何学好这一课程?怎样才算学好了?下面从运用计算机解决实际问题的过程来谈谈本课程在软件设计中的作用。

1. 用计算机解决实际问题的过程

在用计算机解决一个实际问题时,通常涉及到以下几个步骤:

(1) 建立模型

一般情况下,程序设计中需要解决的实际应用问题可能会是各式各样的。例如,我们所熟悉的工资表的处理问题,学生成绩管理问题,电话号码查询问题,从一组数据中选择若干元素的排列、组合,在国际象棋棋盘上放置八个皇后,以使相互间不被吃掉等。这些问题无论是所涉及到的数据还是其操作要求可能都存在一定的差异。尽管如此,许多应用问题之间还是具有一定的相似之处的。例如,虽然工资表和学生成绩表的具体信息(栏目)不同,但如果将两个表中的每个人的工资信息和成绩信息看做一个整体,则这两个表结构之间就具有了某些共性。另外,

从操作方面来看,虽然两个表的操作存在差异,但也一定存在一些相同的基本操作,例如,查询一个人的工资信息和成绩信息,修改有关信息等。类似的例子有很多。

正因为许多不同的问题之间存在着某些共性,使得我们可以将一个具体的问题用这些共性的形式描述出来,这就是通常所说的**建立模型**。建立问题的模型通常包括问题中的数据对象及其关系的描述、问题求解的要求及方法等方面。建立问题模型的优点是:因为所涉及到的许多基本模型在有关的课程中已有介绍,因而通过建立模型,就可以将一个具体的问题转换为所熟悉的模型,然后借助于这一模型来实现。数据结构、离散数学及许多数学课程中就介绍了许多模型。例如,要描述一个群体中个体之间的关系时,我们可以采用数据结构和离散数学中所介绍的图结构;要描述一个工程内的关系或进展情况时,可以采用数据结构所介绍的 AOV 网或 AOE 网等。

(2) 构造求解算法

在建立了模型之后,一个具体的问题就转变成了一个用模型所描述的抽象的问题。借助于这一模型以及已有的知识(例如数据结构中有关图结构的基本知识),我们可以相对容易地描述出原问题的求解方法——算法。从某种意义上说,该算法不仅能实现原问题的求解,而且还可能实现许多类似的具体问题的求解,尽管这些具体问题的背景及其描述形式可能存在较大的差异。

(3) 选择存储结构

在构造出求解算法之后,就需要考虑在计算机上实现求解了。为此,首先要选择合适的存储结构,以便将问题所涉及到的数据(包括数据中的基本对象及对象之间的关系)存储到计算机中。不同的存储形式对实现问题的求解有较大的影响,所占用的存储空间也可能有较大的差异。

(4) 编写程序

选择了存储结构之后,就可以编写程序了。存储形式和问题要求决定了编写程序的方法。

(5) 测试

在编写出完整的程序之后,需要经过测试才能交付使用。

2. 数据结构课程在软件设计中的作用

在介绍了用计算机求解问题的过程之后,下面我们来看看数据结构课程与此有什么关系?事实上,本课程涉及到上述求解过程中的大多数步骤。

(1) 与建立模型的关系

数据结构课程中介绍了许多基本的数据结构模型及其运算实现。例如,线性表、栈和队列、树和二叉树、图、二叉排序树、堆等。通过学习,不仅可以掌握这些基本内容及其应用,还能根据实际问题选择合适的模型。

(2) 与算法设计的关系

课程中对每种数据结构都讨论了相应的基本运算的实现,并且其中的一些算法是非常经典的,掌握这些基本运算的实现方法有助于进行更为复杂的算法设计。

(3) 与选择存储结构的关系

课程中对每种数据结构都讨论了其具体存储结构及存储结构对运算实现的影响。例如,在第二章中所介绍的对顺序表做插入和删除运算,平均需要移动表中一半的元素,而采用链表结构则不需要移动元素。通过对这些内容的学习和比较,可使学生熟练地选择合理的存储结构。

(4) 与编程之间的关系

在进行各种数据结构的算法程序编写时,涉及到许多具有代表性的程序设计方法。通过对这些方法的学习有助于编程技术的提高。

综上所述,数据结构课程对提高软件设计水平有较大的影响。这一课程在计算机专业课程中具有极其重要的作用,几乎绝大多数学校和研究单位的计算机专业研究生入学考试都将这一课程定为考试课程之一。

1.2.2 数据结构课程的主要内容和特点

数据结构课程是计算机专业重要的专业技术基础课程。在该课程中,要介绍软件设计中常用的基本技术,主要包括软件设计中所涉及到的各种数据结构、存储实现及其运算等。通过这些内容的学习,使学习者不仅能掌握这些数据结构的组织形式,而且还能掌握在各种存储形式下实现运算的方法以及各种存储形式对运算的性能特点的影响。由此可见,熟练掌握本课程所介绍的内容,不仅可以有效地提高程序设计的技术,而且对后续课程的学习有很大的帮助。

然而,本课程无论从内容上还是难度上,学习它都需要花费较多的精力,这是因为:

(1) 由于有多种数据结构,每种数据结构又可能有多种存储形式,而采用不同的存储形式时运算的实现可能存在较大的差异,因此,需要掌握的内容较多。

(2) 因为课程中所涉及到的许多递归(递归定义和递归调用)以及动态变量是初学者所难以掌握的,因此,学习难度较大。

由此可知,掌握适合于本课程的学习方法是非常必要的。

1.2.3 数据结构课程的学习方法

如前所述,本课程需要掌握的内容较多,范围广,而且还具有一定的难度,因此需要有合适的方法。下面就有关的学习方法谈谈个人的观点供大家参考。

(1) 了解本科段与专科段之间的差异

在目前通常的教学中,计算机专科层次侧重于计算机软硬件系统的应用和开发,因而要求学生要掌握计算机软硬件课程的基础知识和应用。从自学考试大纲的要求来看,侧重于各门课程中知识的掌握。就数据结构课程而言,不仅要熟练掌握考试大纲中列举的各知识点,还要深刻领会各种算法思想,这样才能对给定的实际问题综合运用所学知识进行求解,设计出符合要求的数据结构和算法。

(2) 正确理解考试大纲

考试大纲是考试、命题的依据,因此考生要严格按照大纲的要求进行学习。通过阅读大纲,可以了解到课程的性质、任务、重点、难点及与其他课程之间的关系,从而做好必要的准备,为后面的学习奠定基础。对照大纲,考生不仅可以在学习时知道自己该学什么,学到什么程度,各项内容的分解等,而且还可以在复习时作为复习提纲和标准,以找出自己学习中存在的问题。因此,考试大纲是整个学习阶段必不可少的学习工具。

(3) 掌握程序设计语言

按照自学考试所指定的教材,本课程主要介绍各种数据结构及其运算实现,因此要求考生必须掌握描述数据结构和算法的工具:程序设计语言及编程技术,即需要掌握 C 语言。

(4) 注意学习内容的联系

虽然本课程的学习内容较多,但如果组织合理,则可以较方便地掌握。在学习每种数据结构时,特别是在复习时,应注意以下几个方面的内容及其联系:

- 这种数据结构的逻辑定义及相关的术语。
- 在这种数据结构上所定义的运算。
- 如何用程序设计语言中的数据结构来表示这种数据结构?
- 在所选用的存储结构上,如何实现所定义的运算?
- 在选定的存储结构上所实现的各算法的性能如何?(主要是时间性能,在有些情况下还要涉及到空间性能。)

(5) 注意实践

考生在学习过程中应做大量的习题,以理解和巩固所学的内容。在编写算法时,应能根据实际问题的要求提出尽可能合理的数据结构,设计出有效的算法,并注意一题多解及比较各解之间的性能。这样既可以巩固所学内容,又能提高解决实际问题的能力,因而是必不可少的重要环节。

由于本课程中算法的执行过程较复杂,因此按传统方法阅读算法有一定的困难,这给验证算法的正确性增加了难度,从而影响到学习的效果。为此,建议有条件的考生尽可能多地在机器上验证算法。

(6) 及时复习和总结

每当一个完整的内容(指每一节、每一章及整个课程)学习完成时,要注意及时复习和总结。复习时,可按大纲所列出的要点进行。在学完每一章、每一节后,应按上述的线索做总结对比,学完全书后还要做全面系统的总结。

1.3 本章内容学习和辅导

1.3.1 有关概念的理解

在教材第一章中介绍了许多概念,由于概念较多,并且对初学者来说较抽象,因而不易理解。下面用归纳总结的方法介绍这些概念。

如教材所述,数据是指信息的载体,能被计算机识别、存储和加工处理。数据的形式较多,例如前面所述的工资报表、学生成绩表,一个家族关系的表示形式,表示一个群体中个体之间关系的图形描述等,如图 1-1(a)~(d)所示。

虽然这些数据的形式及运算存在较大的差异,但可以找出其中的共性部分:每一数据都是由多个具有独立意义的个体所组成,每一数据中的个体之间存在着某些关系,对这些数据的运算也有一些相似部分。例如,在家族关系数据中,组成数据的基本个体是每个人,人与人之间存在着多种关系,例如父子关系、兄弟关系、祖先-后代关系等,其中有些关系是直接表示出来的,还有一些关系则是隐含的。对家族关系数据,通常要涉及到查询特定个体间的关系,插入和删除个体等。其他数据具有相似的组成,及包含多个成员,成员间具有一定的关系,对这些数据需

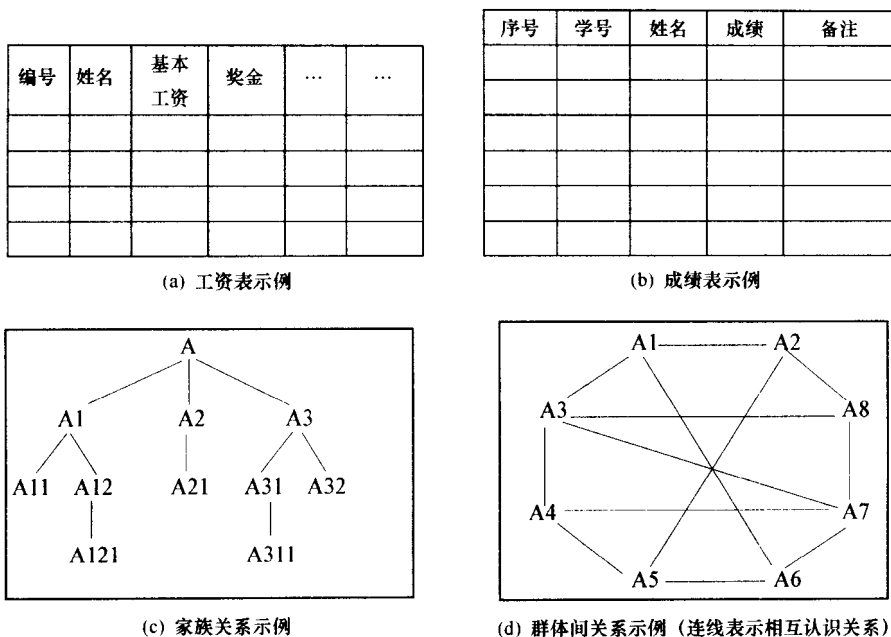


图 1-1 数据示例

要进行特定的运算。

数据结构课程主要讨论这些具有共性的内容。为便于讨论,将这些共性部分分别给出有关描述,得到下面的概念:

(1) 数据元素

数据中具有独立意义的个体。例如工资表中的每个人,成绩表中的每个学生,家族关系中的每个人等。在有些场合下,也称之为**元素**、**记录**、**结点**、**顶点**等。

(2) 字段(域)

虽然将具有独立意义的个体用元素来表示,但在许多情况下还需要特定个体的具体信息,因而涉及到元素的字段信息。字段是对元素的详细描述,通常情况下,元素包含多个字段信息。例如,在图 1-1 所示的成绩表中,每个元素包括序号、学号、姓名、成绩、备注 5 个字段,分别描述了每个学生在表中的序号,学生的学号、姓名、成绩及备注信息。

(3) 数据结构

数据结构是指组成数据的元素之间的结构关系。在图 1-1 所示的几个数据示例中,元素之间各自具有一定的构成形式,这些构成形式被称为数据结构。在图 1-1 中,(a)、(b)中的元素依次排列,构成**线性结构**;(c)中的元素是按树的形式构成**树型结构**;(d)中的元素构成**图结构**。线性结构、树型结构和图结构是数据结构课程中要介绍的几类常见的数据结构。如果数据中的元素之间没有关系,则构成**集合**,这也是一种结构。

通常,称这几类结构为**逻辑结构**,因为仅考虑了元素之间的逻辑关系,而没有考虑其在计算机中的具体实现。

在用计算机解决涉及到数据结构的问题时,需要完成如下任务:

① 数据结构的存储：为所涉及到的数据结构选择一种存储形式，并将其存储到计算机中，这就是数据结构在内存中的**存储结构**（有时也称为**物理结构**）。在教材的后面部分，你将看到每种逻辑结构可能会有多种存储结构。

② 数据结构**运算**的实现：在选择了一种数据结构的存储结构之后，就可以实现所给出的运算了。在数据结构课程中，运算的实现一般是以算法的形式给出的。在以 C 语言作为算法描述语言的教材中，算法以函数的形式给出。

由此可见，对一种数据结构，需要涉及到其逻辑结构、存储结构和运算三个方面。在以后的学习中，对每种结构都要注意这三方面的联系。为便于学习，下面将这三个方面展开，如图 1-2 所示。

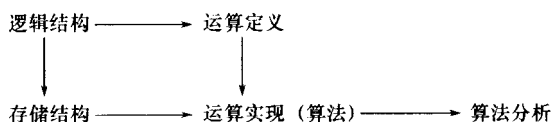


图 1-2 数据结构的组成及联系

由图 1-2 可知，逻辑结构是由定义给出的，运算则取决于逻辑结构；在选择了一种存储结构之后，需要在该存储结构上实现所给出的运算，从而得到了算法。

由于不同的存储形式对算法的时间性能、空间性能等有较大影响，即使是相同的存储结构也可能存在不同的算法实现，为此，需要解决这样的问题：究竟是何种存储结构更为合适？什么算法更有效？为此，需要对算法进行分析，这就是图 1-2 中最后所列出的内容，这将在后面具体介绍。通过分析，可以知道所实现的算法的性能及所选择的存储结构是否符合要求。

1.3.2 算法描述及分析

如前所述，对数据结构的运算是以算法的形式来描述的。算法的定义在教材中已经给出，因而此处不再讨论。读者应理解算法就是一段程序，该程序段对给定的输入可在有限的时间内产生出确定的输出结果。

1. 算法描述语言

算法可采用多种描述语言来描述，例如自然语言、计算机语言或某些伪语言。各种描述语言对问题的描述能力是存在差异的。例如，自然语言较为灵活，但不够严谨，而计算机语言虽然严格，但由于语法方面的限制，使得灵活性不足，因此许多教材中采用的是以一种计算机语言为基础，适当添加某些功能或放宽某些限制而得到的一种类语言，这些类语言既具有计算机语言的严格性，又具有其灵活性，同时也容易上机实现，因此被广泛接受。目前的许多数据结构教材就是采用类 PASCAL、类 C++ 或类 C 语言作为算法描述语言。本教材选用 C 语言作为算法描述语言，所涉及到的算法都是以 C 语言的函数形式给出的。为充分理解教材中的算法，进而能熟练地写出问题的求解算法，要求读者熟练掌握所采用的描述语言。

2. 算法分析

如前所述,为了掌握算法的有关性能,需要对算法进行分析。通过分析,不仅可以知道算法的有关性能,而且还可知所选择的存储结构是否符合要求。

在数据结构课程中,大多数算法分析是针对算法的时间性能进行的。算法的时间性能是以时间复杂度来衡量的,此处所谓算法的**时间复杂度**是指运行算法所需的时间。显然,为了使算法的时间复杂度便于比较,不宜采用在某个具体机器上运行算法所需的时间的形式来表示,对此,一般是以算法中**基本语句的执行次数**来衡量的。然而,在实际应用时,基本语句的执行次数的精确计算是困难的,同时也是不必要的,因为许多算法中的语句的执行次数要取决于输入数据,可能会有多种复杂的情况。为便于计算,对这一时间复杂度大多采用一种近似的形式来描述,即采用基本语句执行次数的**数量级**来表示。此处所谓数量级是这样定义的:

如果变量 n 的函数 $f(n)$ 和 $g(n)$ 满足: $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \text{常数 } k (k \neq \infty, 0)$, 则称 $f(n)$ 和 $g(n)$ 是同一数量级的,并用 $f(n) = O(g(n))$ 的形式来表示。

由定义可知,两个函数为同一数量级,强调的是在 n 趋向无穷大时,两者“接近”。例如,

$\frac{n(n+1)}{2}$ 和 n^2 是同一数量级的,因为 $\lim_{n \rightarrow \infty} \frac{\frac{n(n+1)}{2}}{n^2} = 1/2$ 为一个常数。

通过求解算法中基本语句执行次数的数量级所得到的时间复杂度忽略了其中的“细微”部分,得到了时间性能的近似描述,这一描述有助于对算法时间性能的简要了解。

例 1.1 求解下列各程序段的时间复杂度。

(1) for ($I = 1; I < n; I++$) { $x++$; }

解答: 虽然从循环语句的内部执行来看,该程序段中基本语句的执行次数多于循环体的执行次数,但两者是同一数量级的,因此,该问题的求解可以以其循环体的执行次数的数量级的求解来实现。由于循环体执行 $n-1$ 次,因此其时间复杂度为 $O(n)$ 。

(2) for ($I = 1; I < n; I++$)

for ($j = 1; j <= I; j++$) { $x++$; }

解答: 该问题同样以其循环体的执行次数的数量级的求解来实现。由于是双重循环,并且内层循环的循环次数不是常数,因此其计算稍复杂。计算方法如下:

$I = 1$ 时,内层循环执行 1 次(j 从 1 到 1);

$I = 2$ 时,内层循环执行 2 次(j 从 1 到 2);

$I = 3$ 时,内层循环执行 3 次(j 从 1 到 3);

...

$I = n-1$ 时,内层循环执行 $n-1$ 次(j 从 1 到 $n-1$);

因此,最内层循环体共执行 $n(n-1)/2$ 次,因此其时间复杂度为 $O(n^2)$ 。

(3) $I = 1$;

while ($I < n$) $I * = 2$;

解答: 对许多初学者来说,该题的求解比较复杂,求解方法如下:

由语句可知,循环次数和 I 之间的对应关系如下:

次数	1	2	3	4	...	...	k
I 值	2^1	2^2	2^3	2^4	...	...	2^k

由此可知,该语句段的时间复杂度为 $O(\log_2 n)$ 。

1.4 复 习 题

1. 给出一些数据的例子,指出其中的元素、元素的字段、元素之间的关系,并给出一些最基本的运算。

2. 已知一个群体中有 n 个人,这些人之间可能存在同学关系,请用一个数据模型来描述这一关系,并给出可能的基本运算。

3. 计算下列各程序段的时间复杂度。

① for ($I=0; I < n; I++$)

 for ($j=I; j < n; j++$) $x++$;

② $I=n$;

 while ($I > 1$) $I /= 2$;

③ for ($I=1; I < n; I++$)

 for ($j=1; j < n; j++$)

 for ($k=1; k < n; k++$)

$x++$;

④ for ($I=1; I < n; I++$)

 for ($j=1; j < n; j++$) $x++$;

 for ($k=1; k < n; k++$) $x++$;

⑤ for ($I=1; I < n; I++$)

 { $j=I$;

 while ($j < n$) $j *= 2$;

 }

第二章 线 性 表

2.1 本章内容概述及学习方法

线性表是数据结构课程中介绍的第一种数据结构,这是一种最基本的数据结构,是软件设计中最为常用的一种数据结构。本章是整个数据结构课程的基础,真正学好本章有利于整个课程的学习,因此,在某种程度上说是本课程学习成败的关键。下面首先系统地介绍本章的有关知识,然后再给出其应用示例。

在学习本章前,应掌握算法描述语言,特别是其中有关指针和动态变量的内容。在学习过程中,应注意在系统掌握基本概念的基础上着重理解算法的思想、性能,并能灵活应用于实际问题的求解。本辅导书在每一章节讲解有关内容之后列出一些思考题,用来帮助考生及时发现学习中存在的问题,所以请认真思考。

2.1.1 内容简介

教材在这一章中,首先介绍了线性表的**逻辑结构、运算描述**以及有关的术语。

在选择线性表的存储结构时,首先讨论了采用顺序存储方式存储线性表,从而得到了**顺序表结构**。在顺序表结构中,逻辑上相邻元素的存储地址也相邻。通过在顺序表结构上实现所给出的基本运算及算法分析可知,在顺序表结构上实现插入和删除运算时需要移动较多的元素(在等概率情况下,插入或删除一个元素平均需要移动一半的元素),因而提出了线性表的**链式存储结构,即链表结构**。

在链表结构中,逻辑上相邻的元素的存储位置不一定相邻,元素之间的逻辑次序是通过**指针(链)**来描述的。为了运算及描述的一致性,在链表中设置了一个**头结点**,从而得到带头结点的单链表结构形式。在(带头结点的)单链表上实现各种基本运算是本章的重要内容,许多更为复杂的运算都是建立在对这些基本运算的真正理解的基础之上的。

在有些情况下,为了实现特定问题的求解,需要将链表的首尾结点相连接,即让尾结点的后继指针指示表头结点,从而得到循环链表结构。在循环链表中的运算实现与前面链表结构上的运算实现基本类似,所不同的是,循环链表中尾结点的判断显然不同,因此要求算法中能注意到这一点,以免出现“死循环”。对某些特定问题,例如要实现链表的首尾相接,需要能比较方便地知道链表的首尾结点,为此可采用带尾指针的单循环链表形式。

在前面所讨论的单链表中,不能较方便地求出每个结点的前趋结点。如果需要通过实现这一功能的话,可在每个结点中再增设一个指示其前趋的指针,从而得到双(向)链表结构。双链表也可以带头结点,也可以设置为循环形式。在双链表上的大多数运算实现与单链表上的运算实现有许多相似之处,明显不同的是插入和删除结点的运算。

在各种链表结构上搜索结点及插入和删除运算是最基本的运算要求,因为这是复杂运算的