高等学校教材

数据结构

——C语言描述

耿国华　主编

高等教育出版社

高等学校教材
数据结构——C语言描述
耿国华　主编
高等教育出版社

内容提要

本书共分10章，内容包括基本概念、基本结构（线性表、栈和队列、串、数组与广义表、树、图）和基本技术（查找方法与排序方法）三大部分，其中贯穿了程序设计中参数传递技术、动态处理的指针技术、数组技术、递归技术与队列等技术。书中给出了许多经典算法，每章均附有小结与典型题例，便于总结提高。附录给出了2套学期考题样卷和4套硕士研究生入学考试的样卷，便于读者模拟练习和考研参考。

本书集作者多年教学实践经验，采用面向对象的方法讲述数据结构技术，用标准C描述算法，内容丰富，概念清楚，技术实用。课程教学资源丰富是本书的特色，配套光盘包括部分扩展内容、课程设计与课程实习指导、多媒体教学课件、算法程序示例和算法转换为程序的模板，本书的PPT电子教案可从高等教育出版社网站http://www.hep.edu.cn免费下载，更多资源请浏览陕西省精品课程网站和西北大学精品课程网站http://jpkc.nwu.edu.cn。

本书可作为高等学校计算机及相关专业数据结构课程的教材，也可供从事计算机应用开发的工程技术人员参考使用。

图书在版编目(CIP)数据

数据结构：C语言描述/耿国华主编. —北京：高等教育出版社，2005.7
ISBN 7-04-016457-4

Ⅰ.数… Ⅱ.耿… Ⅲ.①数据结构②C语言-程序设计 Ⅳ.①TP311.12②TP312

中国版本图书馆CIP数据核字(2005)第067511号

出版发行	高等教育出版社	购书热线	010-58581118
社　　址	北京市西城区德外大街4号	免费咨询	800-810-0598
邮政编码	100011	网　　址	http://www.hep.edu.cn
总　　机	010-58581000		http://www.hep.com.cn
		网上订购	http://www.landraco.com
经　　销	北京蓝色畅想图书发行有限公司		http://www.landraco.com.cn
印　　刷	北京铭成印刷有限公司		
开　　本	787×1092　1/16	版　　次	2005年7月第1版
印　　张	23.75	印　　次	2005年7月第1次印刷
字　　数	510 000	定　　价	28.00元（含光盘）

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。

物料号　16457-00

前　言

我们生活在一个物质的世界，计算机工作者又面对着数字的世界，如果将物质世界中的事与物数字化，那么它们在计算机中的表现均为数据。这些数据来源于现实，表征着具体的意义，而且在计算机中有着统一的表示方法，因而成为被计算机程序处理的符号集合。研究数据在计算机中的表示方法、关联方法、存储方法以及在其上的典型处理方法，就构成了数据结构课程的主要内容。

由于数据是计算机处理的对象，使用计算机的过程就是对数据进行加工处理的过程，因而数据的组织与结构被确立为计算机科学中最为基本的内容。早在20世纪80年代初，数据结构课程就已成为国内计算机专业教学计划中的核心课程。IEEE－CS/ACM的CC2001教程也将算法与数据结构课程列为核心课程之首，数据结构愈显出其在信息学科中的重要地位。

人类解决问题的思维方式可分为推理方式和算法方式两大类。推理方式凭借公理系统思维方法，从抽象公理体系出发，通过演绎、归纳、推理求证结果，解决特定问题。这种推理方式是通过数学训练得到的。算法方式则是凭借构造性思维，从具体操作规范入手，通过操作过程的构造实施来解决特定问题。

数据结构的学习过程，是进行复杂程序设计的训练过程，是算法构造性思维方法的训练过程，技能培养的重要程度不亚于知识传授。本门课程教学的重要内容和主要难点在于让学生理解、习惯算法构造思维方法。培养学生的数据抽象能力、算法设计能力以及创造性思维方法，才能够举一反三、触类旁通，从而达到应用知识解决复杂问题的目的。

数据结构作为专业基础课程，一般开设在大学二年级，应对前两年学习的软件技术进行总结提高，为后续专业课程提供基础，它承上启下，贯通始终，是计算机科学与技术人才素质框架中的脊梁，对学生能力培养至关重要。通过对数据结构的学习，读者能够以问题求解方法、程序设计方法及一些典型的数据结构算法为研究对象，学会分析数据对象特征，掌握数据组织方法和在计算机中的表示方法，为数据选择适当的逻辑结构、存储结构以及相应的处理算法，初步掌握算法的时间、空间复杂度分析基础，培养良好的程序设计风格以及进行复杂程序设计的技能。

本书以抽象数据类型为中心，采用面向对象的新观点，将教学内容分为基本概念、基本结构、基本技术三个层次，并贯穿了计算机科学中的一些重要的问题求解技术，符合认知规律。使用熟悉的标准C作为算法描述的语言，使之与大多数院校讲授的第一语言衔接，便于将读者的注意力集中在算法的理解上。书中给出了用C语言实现抽象数据类型的方法，大量的C函数的程序实例正是数据抽象与过程抽象的结合。这就使数据结构的表示得以简

化,突出了算法表示的实质。

课程教学资源丰富是本书的特色。鉴于数据结构课程的技术性特点,我们建设了与教材配套的多媒体课件和课程教学网站,提供师生、学生及同行之间的课程教学资源与交流平台。随书所附光盘包括面向教学的多媒体课件;面向学习的算法 C 语言实现和课程设计与实习范例;面向扩展的补充学习内容和算法转换为程序的模板示例。多媒体课件突出数据结构课程难点算法本质的理解,采用 Flash 动画实现全部算法执行过程,便于学生从直观的感性认识过渡到理性的深入理解,丰富了学生获取信息的种类、方式;课件提供的选件控制方式,有助于教师形象生动、深入浅出地讲解,便于对授课内容重新组合,实现教学过程的个性化;为方便学习和实践,光盘中所列算法函数只需引用相应的类型定义头文件就可直接上机运行;给出课程实习和设计模板,强化实习和设计核心内容。PPT 电子教案可从高等教育出版社网站上免费下载,其他教学资源可访问陕西省精品课程网站和西北大学精品课程网站。

本书共分 10 章,包括三大部分,其中第一部分(第 1 章)是数据结构的基本概念部分;第二部分(2 ~ 7 章)是基本的数据结构部分,包括线性结构(线性表、栈和队列、串、数组与广义表)与非线性结构(树、图);第三部分(8 ~ 10 章)是基本技术部分,包括查找方法与排序方法。其中还贯穿了一些重要的程序设计技术,如参数传递技术、动态处理的指针技术、数组技术(抽象规律处理)、递归技术与队列技术;书中给出了许多精彩的典型算法,是人们在数据处理中智慧的结晶,我们力求将经典算法的思路表现出来,为学习者继续拓展提供线索。本书每章均附有小结与典型题例,便于总结提高。在附录中考题部分给出了两套学期考题样卷,其余 4 套样卷均为硕士研究生入学考试的样卷,以便读者练习。鉴于篇幅将部分选学内容放入光盘。

曾指导我学习数据结构的清华大学严蔚敏教授和唐泽圣教授,他们敏锐的洞察力和对教学内容的精辟讲解,包括尔后多年中所给予的多方面指导,使我受益终生。在此表示衷心的感谢!

本书由耿国华教授任主编,张德同副教授任副主编。其中的第 1 章、6 章、7 章、9 章及附录由耿国华编写;第 5 章、8 章及实验指导由张德同编写;第 3 章、10 章由周明全编写;第 4 章由冯宏伟编写,第 2 章由卢燕宁编写。全书由耿国华统稿。本书算法均在 TURBO C2.0 环境下调试通过。李康负责教学网站建设,多名研究生参加算法调试与动画多媒体课件制作。在本书中,我们将多年从事数据结构课程教学的体会写了出来,恳请读者赐教指正。

耿国华

2005 年 6 月

目　录

第1章 绪 论

陈火旺院士把计算机50多年的成就概括为五个“一”:开辟一个新时代——信息时代;形成一个新产业——信息产业;产生一个新学科——计算机科学与技术;开创一种新的科研方法——计算方法;开辟一种新文化——计算机文化。这一概括深刻阐明了计算机对社会发展广泛而深远的影响。

数据结构被称为是计算机科学的两大支柱之一。著名的计算机科学家P. Wegner指出:“在工业革命中起核心作用的是能量,而在计算机革命中起核心作用的是信息”。计算机科学就是“一种关于信息结构转换的科学”。

关于数据结构理论的研究,可以追溯到1972年C. A. R. Hoare奠基性的论文《数据结构笔记》;而现代计算机所大量采用的各种数据结构,最早的系统论述应归于D. E. Knuth的名著《计算机程序设计技巧》。随着计算机科学的飞速发展,数据结构的基础研究也逐渐走向成熟。

计算机科学是关于信息结构转换的科学,信息结构(数据结构)应当是计算机科学研究的基本课题。计算机科学的重要基石是关于算法的学问,数据结构又是算法研究的基础。

开始数据结构课程之前,我们需要在绪论中回答以下问题:

(1) 定义(什么是数据结构);

(2) 内容(数据结构的研究范围);

(3) 方法(研究采用的方法);

(4) 描述(算法规则描述的工具);

(5) 评价(对算法作性能评价);

(6) C语言(工具要点、规范要求);

(7) 关于数据结构的学习。

本章将通过对这些问题与概念的简要介绍,描述数据结构基本内容、主要概念与描述规范,作为本门课程的梗概之序。

1.1 数据结构的基础概念

首先介绍数据结构的相关名词。

(1) 数据(Data)

数据是描述客观事物的数值、字符以及能输入机器且能被处理的各种符号集合。换句话说,数据是对客观事物采用计算机能够识别、存储和处理的形式所进行的描述。简而言之,数据就是计算机化的信息。

数据概念经历了与计算机发展相类似的发展过程。计算机一问世,数据作为程序的处理对象随之产生。早期计算机主要应用于数值计算,数据量小且结构简单,数据仅有进行算术运算与逻辑运算的需求,数据只包括整型、实型、布尔型,那时程序设计人员把主要精力放在程序设计的技巧上,并不重视如何在计算机中组织数据。

随着计算机软件、硬件的发展与应用领域的不断扩大,计算机应用领域发生了战略性转移,非数值运算处理所占的比例越来越大,现在几乎达到90%以上,数据的概念被大大推广了。数据包含数值、字符、声音、图像等一切可以输入到计算机中的符号集合,多种信息通过编码被归到数据的范畴,大量复杂的非数值数据要处理,数据的组织显得越来越重要。20世纪70年代后,微型机的普及以及数据库、人工智能的研究推动了计算机技术的发展,人们越来越重视运用科学工具来探索数据和程序的内部关系以及它们之间的关系,采用新的观点来设计计算机体系,使计算技术发展为一门科学。

数据的概念不再是狭义的,数据已由纯粹的数值概念发展到图像、字符、声音等各种符号。

例如,对于C源程序,数据概念不仅是源程序所处理的数据。对于编译程序来说,它加工的数据是字符流的源程序(.c),输出的结果是目标程序(.obj);对于链接程序来说,它加工的数据是目标程序(.obj),输出的结果是可执行程序(.exe),如图1.1所示。

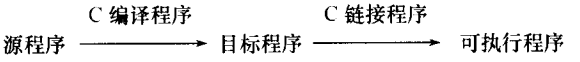

图1.1　编译程序示意图

而对于C编译程序,由于它在操作系统控制下接受操作系统的调度,因此相对操作系统来说它又是数据。

(2) 数据元素(Data Element)

数据元素是组成数据的基本单位,是数据集合的个体,在计算机中通常作为一个整体进行考虑和处理。一个数据元素可由一个或多个数据项组成,数据项(Data Item)是有独立含义的最小单位,此时的数据元素通常称为记录(Record)。如表1-1所示,学籍表是数据,每一个学生的记录就是一个数据元素。

表 1-1 学籍表

数据项

学 号	姓 名	性 别	籍 贯	出生年月	住 址
101	赵虹玲	女	河北	1983.11	北京
⋮	⋮	⋮	⋮	⋮	⋮

← 记录

(3) 数据对象(Data Object)

数据对象是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N=\{0,\pm1,\pm2,\cdots\}$,字母字符数据对象是集合 $C=\{'A','B',\cdots,'Z'\}$,表 1-1 中所示的学籍表也可看做一个数据对象。由此可看出,不论数据元素集合是无限集(如整数集),是有限集(如字符集),还是由多个数据项组成的复合数据元素(如学籍表),只要性质相同,都是同一个数据对象。

综上(1)~(3)所述,再分析数据概念:

其一:数据特点
- 可放入机器(与机器的关联性)
- 可被加工 (能被处理)

其二:数据构成
- 数据元素——组成数据基本单位 (与数据的关系是集合的个体)
- 数据对象——性质相同的数据元素的集合 (与数据的关系是集合的子集)

(4) 数据结构(Data Structure)

数据结构是指相互之间存在一种或多种特定关系的数据元素集合,是带有结构的数据元素的集合,它指的是数据元素之间的相互关系,即数据的组织形式。由此可见,计算机所处理的数据并不是数据的杂乱堆积,而是具有内在联系的数据集合,如表结构(表 1-1 所示的学籍表)、树形结构(如图 1.2 所示的学校组织层次结构图)、图结构(如图 1.3 所示的交通流量图)。我们关心的是数据元素之间的相互关系与组织方式,以及对其施加的运算及运算规则,并不涉及数据元素的内容具体是什么值。

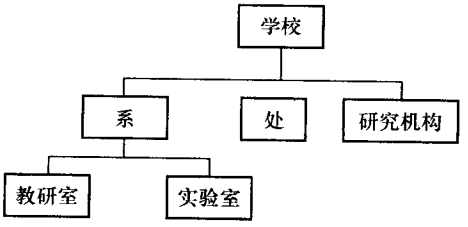

图 1.2 学校组织层次结构图

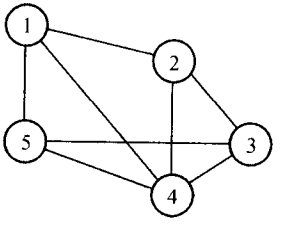

图 1.3 交通流量图

例如,一维数组是向量 $\boldsymbol{A}=(a_1,\cdots,a_n)$ 的存储映像,使用时采用下标变量 A[i]的方式,关注其按序排列、按行存储的特性,并不关心 A[i]中存放的具体值。同理,二维数组 A[i,j]是矩阵 $\boldsymbol{A}_{m\times n}$ 的存储映像,我们关心结构关系的特性而不涉及其数组元素本身的内容。

(5) 数据类型 (Data Type)

数据类型是一组性质相同的值集合以及定义在这个值集合上的一组操作的总称。数据类型中定义了两个集合,即该类型的取值范围以及该类型中可允许使用的一组运算。例如,高级语言中的数据类型就是已经实现的数据结构的实例。从这个意义上讲,数据类型是高级语言中允许的变量种类,是程序语言中已经实现的数据结构(即程序中允许出现的数据形式)。在高级语言中,整型类型可能的取值范围是 -32 767 ~ +32 768,可用的运算符集合为加、减、乘、除、取模(如 C 语言中 +、-、*、/、%)。

从硬件的角度来看,它们的实现涉及到"字"、"字节"、"位"、"位运算"等;从用户观点来看,并不需要了解整数在计算机内是如何表示、运算细节是如何实现的,用户只需要了解整数运算的外部运算特性,而不必了解机器内部位运算的细节,就可运用高级语言进行程序设计。引入数据类型的目的,从硬件的角度是将其作为解释计算机内存中信息含义的一种手段,对使用数据类型的用户来说则实现了信息隐蔽,将一切用户不必关心的细节封装在类型中。如两整数求和问题,用户只注重其数学求和的抽象特性,而不必关心加法运算涉及的内部位运算如何实现。

按"值"的不同特性,一般说来,高级程序语言中的数据类型可分为两大类:一类是非结构的原子类型,原子类型的值是不可分解的,如 C 语言中的标准类型(整型、实型和字符型)及指针;另一类是结构类型,结构类型的值是由若干成分按某种结构组成的,因此是可以分解的,并且它的成分可以是非结构的,也可以是结构的。例如,数组的值由若干分量组成,每个分量可以是整数,也可以是数组等。数据类型指由系统定义的、用户可直接使用且可构造的数据类型。

思考题:C 语言中的指针类型属于原子类型还是结构类型?

(6) 抽象数据类型 (Abstract Data Type,简称 ADT)

抽象的本质是抽取反映问题的本质点,忽视非本质的细节,这正是从事计算机研究的本质。

① 数据的抽象

计算机中使用的是二进制数,汇编语言中则可给出各种数据的十进制表示,如 98.65、9.6E3 等,它们是二进制数据的抽象;程序设计人员在编程时可以直接使用,不必考虑实现细节。在高级语言中,则给出更高一级的数据抽象,出现了数据类型,如整型、实型、字符型等。到抽象数据类型出现,可以进一步定义更高级的数据抽象,如各种表、队、栈、树、图、窗口、管理器等,这种数据抽象的层次为设计者提供了更有力的手段,使得设计者可以从抽象

的概念出发，从整体考虑，然后自顶向下、逐步展开，最后得到所需结果。可以这样看，高级语言中提供整型、实型、字符、记录、文件、指针等多种数据类型，可以利用这些类型构造出像栈、队列、树、图等复杂的抽象数据类型。

② 抽象数据类型

抽象数据类型定义了一个数据对象、数据对象中各元素间的结构关系以及一组处理数据的操作。从某种意义上讲，抽象数据类型和数据类型实质上是一个概念，只不过 ADT 更为广义，不仅限于各种不同的计算机处理器中已定义并实现的数据类型，还包括设计软件系统时用户自己定义的复杂数据类型。

ADT 包括定义和实现两方面，其中定义是独立于实现的。定义仅给出一个 ADT 的逻辑特性，不必考虑如何在计算机中实现。

抽象数据类型的定义取决于客观存在的一组逻辑特性，而与其在计算机内如何表示和实现无关，即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部使用，从而为实现软件的部件化和可重用性提供了理论保证，进而提高了软件生产率。

ADT 通常是指由用户定义且用以表示应用问题的数据模型，通常由基本的数据类型组成，并包括一组相关服务操作。本门课程中将要学习的表、堆栈、队列、串、树、图等结构就是一个个不同的抽象数据类型。以盖楼为例，直接用砖头、水泥、沙子来盖，不仅建造周期长，且建造高度规模受限。如果用公司按规范提供的水泥预制板，不仅可以高速、安全地建造高楼，水泥预制板使高楼的接缝量大大减少，从而减低了建造高楼的复杂度。由此可见，抽象数据类型是大型软件构造的模块化方法，典型的表、堆栈、队列、串、树、图抽象数据类型，就相当于是设计大型软件的“水泥预制板”，用这些已由专门公司设计好的抽象数据类型就可以安全、快速、方便地设计功能复杂的大型软件。

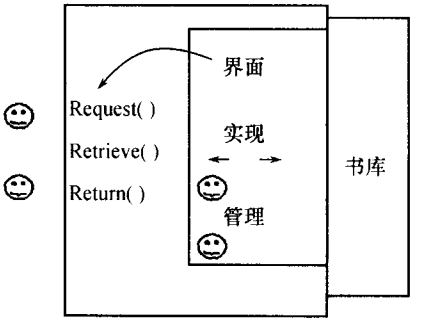

图 1.4 信息隐蔽示意图

抽象数据类型最重要的特点是数据抽象与信息隐蔽，抽象的本质是抽取反映问题的本质点，忽视非本质的细节，从而使设计的数据结构更具一般性，可以解决一类问题。信息隐蔽就是对用户隐藏数据存储和操作实现的细节，使用者仅需了解操作或界面服务，通过使用界面服务来访问数据。如图 1.4 所示：

Request( )：检查书库，回答读者所借图书是否存在；

Retrieve( )：书库取书，送书给读者；

Return( )：放书入库，将读者所还书放入书架。

抽象数据类型的特征是使用与实现分离，实现封装和信息隐蔽，也就是说，在抽象数据类型设计时，类型的定义与其实现分离。

数据类型的抽象层次越高，含有该抽象数据类型的软件复用程度就越高。ADT 定义该抽象数据类型需要包含哪些信息，并根据功能确定公共界面的服务，使用者可以使用公共界面中的服务对该抽象数据类型进行操作。从使用者的角度看，只要了解该抽象数据类型的规格说明，就可以利用其公用界面中的服务来使用这个类型，不必关心其物理实现，从而集中考虑如何解决实际问题。

ADT 物理实现作为私有部分封装在其实现模块内，使用者不能看到，也不能直接操作该类型所存储的数据，只有通过界面中的服务来访问这些数据。从实现者的角度来看，把抽象数据类型的物理实现封装起来，有利于编码、测试，也有利于修改。当需要改进数据结构时，只要界面服务的使用方式不变，只需要改变抽象数据类型的物理实现，所有使用该抽象数据类型的程序不需要改变，这样就会提高系统的稳定性。

1.2　数据结构的内容

在 1.1 节中已给出数据结构一个概念，数据结构是指相互之间存在一种或多种特定关系的数据元素集合。这个描述是一种非常简单的解释。数据元素间的相互关系具体应包括三个方面：数据的逻辑结构、数据的物理结构和数据的运算集合。

（1）逻辑结构

数据的逻辑结构是指数据元素之间的逻辑关系描述。

数据结构的形式定义为：数据结构是一个二元组

$$\text{Data\_Structure} = (D, R)$$

其中：D 是数据元素的有限集，R 是 D 上关系的有限集。

例如，

DS2 = ($D2$, $R2$)

D2 = {a, b, c, d, e, f}

R2 = {T}

T = {〈a, b〉, 〈a, c〉, 〈a, d〉, 〈c, e〉, 〈c, f〉}

则其结构图为一棵树。

根据数据元素之间关系的不同特性，通常有下列**四类基本结构**（如图 1.5 所示）。

① **集合结构**：结构中的数据元素之间除了同属于一个集合的关系外，无任何其他关系。

② **线性结构**：结构中的数据元素之间存在着一对一的线性关系。

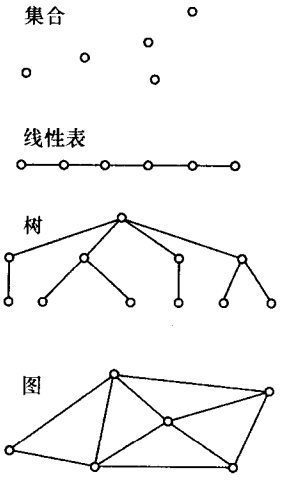

图 1.5　4 类基本数据结构示意图

③ **树形结构**:结构中的数据元素之间存在着一对多的层次关系。

④ **图状结构**或网状结构:结构中的数据元素之间存在着多对多的任意关系。

由于集合的关系非常松散,因此可以用其他结构代替它,故数据的逻辑结构可概括如下:

$$\text{逻辑结构}\begin{cases}\text{线性结构—线性表、栈、队、字符串、数组、广义表}\\ \text{非线性结构—树、图}\end{cases}$$

(2) 存储结构

存储结构(又称物理结构)是逻辑结构在计算机中的存储映像,是逻辑结构在计算机中的实现,它包括数据元素的表示和关系的表示。

形式化描述:D 要存入机器中,建立一个从 D 的数据元素到存储空间 M 单元的映像 S,$D \rightarrow M$,即对于每一个 d, $d \in D$,都有惟一的 $z \in M$ 使 $S(D) = Z$, 同时这个映像必须明显或隐含地体现关系 R。

逻辑结构与存储结构的关系为:存储结构是逻辑关系的映像与元素本身映像。逻辑结构是数据结构的抽象,存储结构是数据结构的实现,两者综合起来建立了数据元素之间的结构关系。

数据元素之间的关系在计算机中有两种不同的表示方法:

- 顺序映像(顺序存储结构);
- 非顺序映像(非顺序存储结构)。

数据结构在计算机中的映像包括数据元素映像和关系映像。关系映像在计算机中可用顺序存储结构或非顺序存储结构这两种不同的表示方法来存放。逻辑结构在计算机存储器中实现时,可采用不同的存储器来表示,不论是内存表示或外存表示,都要以反映逻辑关系为原则。

(3) 运算集合

讨论数据结构的目的是为了在计算机中实现操作,因此在结构上的运算集合是很重要的部分。数据结构就是研究一类数据的表示及其相关的运算操作。

通过下面的工资表(如表 1-2 所示)实例对数据结构的内容作一概括。

表 1-2 工资表

编 号	姓 名	性别	基本工资	工龄工资	应扣工资	实发工资
100001	张爱芬	女	845.67	145.45	30.00	451.12
100002	李 林	男	945.90	185.60	45.00	586.50
100003	刘晓峰	男	945.00	130.00	25.00	450.00
100004	赵 俊	女	860.90	225.90	65.00	721.80
100005	孙 涛	男	950.60	190.80	50.00	591.80
…	…	…	…	…	…	…
100121	张兴强	男	1025.98	365.53	100.00	1291.51

在表1-2所示工资表中采用线性表的逻辑结构：因为结点与结点之间是一种简单的线性关系；存储结构：工资表可包括几千名职工信息，可采用顺序方式存放，也可采用非顺序方式存放。怎么存放就是具体存储结构问题。对于工资表，当职工调离时要删除数据元素，调进时要增加数据元素，调整工资时要修改数据元素。这里的增、删、改就是数据的操作集合。

综上所述，数据结构的内容可归纳为三个部分：逻辑结构、存储结构和运算集合。**按某种逻辑关系组织起来的一批数据，按一定的映像方式把它们存放在计算机的存储器中，并在这些数据上定义一个运算的集合，就叫做数据结构。**

数据结构主要研究怎样合理地组织数据、建立合适的结构、提高执行程序所用的时空效率。数据结构课程不仅讲授数据信息在计算机中的组织和表示方法，同时也重在训练高效地解决复杂问题的能力。

1.3　算法设计

数据结构与算法之间存在着本质联系，在某一类型数据结构上，总要涉及其上施加的运算，而只有通过对定义运算的研究，才能清楚地理解数据结构的定义和作用；在涉及运算时，总要联系到该算法处理的对象和结果的数据。

在本门课程中，我们将大量地遇到算法问题，因为算法联系着数据在计算过程中的组织方式，为了描述实现某种操作，常常需要设计算法，因而算法是研究数据结构的重要途径。

(1) 算法(Algorithm)定义

Algorithm is a finite set of rules which gives a sequence of operation for solving a specific type of problem. (**算法是规则的有限集合，是为解决特定问题而规定的一系列操作。**)

(2) 算法的特性

① 有限性：有限步骤之内正常结束，不能形成无穷循环。

② 确定性：算法中的每一个步骤必须有确定含义，无二义性。

③ 输入：有多个或0个输入。

④ 输出：至少有一个或多个输出。

⑤ 可行性：原则上能精确进行，操作可通过已实现的基本运算执行有限次而完成。

在算法的五大特性中，最基本的是有限性、确定性和可行性。

(3) 算法设计的要求

当用算法来解决某问题时，算法设计要达到的目标是正确、可读、健壮、高效低耗。一个好的算法一般应该具有以下几个基本特征。

① 算法的正确性

算法的正确性是指算法应该满足具体问题的需求。其中“正确”的含义大体上可以分为

四个层次。

a. 程序没有语法错误;

b. 程序对于几组输入数据能够得出满足要求的结果;

c. 程序对于精心选择的典型、苛刻而带有刁难性的几组输入数据能够得出满足要求的结果;

d. 程序对于一切合法的输入数据都能产生满足要求的结果。

对于这四层含义,其中达到第 d 层含义下的正确是极为困难的。一般情况下,以第 c 层含义的正确作为衡量一个程序是否正确的标准。

例如,要求 n 个数的最大值问题,给出示意算法如下:

```
max = 0;
 for (i = 1 ;i <= n ;i ++)
    { scanf("%f", &x);
      if (x > max) max = x;
    }
```

求最大值的算法无语法错误;当输入的 n 个数全为正数时,结果正确,如果输入的 n 个数全为负数,求得的最大值为 0,显然这个结果不对,由这个简单的例子可以说明算法正确性的内涵。

思考题:上面求最大值的算法到底应当算第几层次?是否能算是正确算法?

② 可读性

一个好的算法首先应该便于人们理解和相互交流,其次才是机器可执行。可读性好的算法有助于人们对算法的理解,反之难懂的算法易于隐藏错误且难于调试和修改。

③ 健壮性(鲁棒性)

即对非法输入的抵抗能力。它强调的是,如果输入非法数据,算法应能加以识别并做出处理,而不是产生误动作或陷入瘫痪。

④ 高效率和低存储量

算法的效率通常是指算法的执行时间。对于一个具体问题的解决通常可以有多个算法,执行时间短的算法其效率就高。所谓的存储量需求,是指算法在执行过程中所需要的最大存储空间,这两者都与问题的规模有关。

1.4 算法描述工具

著名的计算机科学家 N. 沃思给出了一个著名的公式:算法 + 数据结构 = 程序,说明数据结构和算法是程序的两大要素,二者相辅相成,缺一不可。

(1) 算法、语言、程序的关系

首先分析数据结构中算法、语言和程序的关系。

① 算法:描述了数据对象的元素之间的关系(包括数据逻辑关系、存储关系描述)。

② 描述算法的工具:算法可用自然语言、框图或高级程序设计语言进行描述。自然语言简单但易于产生二义性,框图直观但不擅长表达数据的组织结构,而高级程序语言则较为准确但又比较严谨。

③ 程序是算法在计算机中的实现(与所用计算机及所用语言有关)

(2) 设计实现算法过程的步骤

- 找出与求解有关的数据元素之间的关系(建立结构关系)。
- 确定在某一数据对象上所施加的运算。
- 考虑数据元素的存储表示。
- 选择描述算法的语言。
- 设计实现求解的算法,并用程序语言加以描述。

(3) 类描述算法语言的选择

高级语言描述算法具有严格准确的优点,但用于描述算法,也有语言细节过多的弱点,为此采用类语言形式。所谓类语言,是指接近于高级语言而又不是严格的高级语言,它具有高级语言的一般语句设施,撇掉语言中的细节,以便把注意力集中在算法处理步骤本身的描述上。

传统的方法是采用 Pascal 语言,由于该语言语法规范严谨,非常适合于数据结构课程教学。在 Windows 环境下出现了一系列功能强大且面向对象的程序开发工具,如 Visual C++、Boland C++、Visual Basic 等。近年来在计算机科学研究、系统开发、教学以及应用开发中,C 语言的使用范围越来越广,C 语言成为计算机专业与非计算机专业必修的高级程序设计语言。C 语言类型丰富,执行效率高,很多学生在学习数据结构课程之前,都已具备了熟悉 C 语言的基础条件,因此本教材采用了标准 C 语言作为描述算法的工具。

为了便于学习者掌握算法的本质,尽量压缩语言描述的细节,在每一部分所使用的结构类型都统一在相应部分的首部统一定义,类型定义不重复,目的是能够简明扼要地描述算法,突出算法的思路,而不拘泥于语言语法的细节。具体上机调试时,可通过包含头文件方式嵌入相关类型定义。

本书中所采用的是 C 语言,个别处使用了对标准 C 语言的一种简化表示,如变量可省略定义部分直接使用,输入/输出函数中可省掉格式类型,以简化表示。

1.5　对算法做性能评价

一种数据结构的优劣是由实现其各种运算的算法具体体现的,对数据结构的分析实质