21世纪

高等院校计算机系列教材

数据结构
—用C语言描述
（第二版）

宁正元　易金聪　编著

中国水利水电出版社
www.waterpub.com.cn

21世纪高等院校计算机系列教材

数据结构——用C语言描述

（第二版）

宁正元　易金聪　编著

中国水利水电出版社

内 容 提 要

本书是根据高校计算机教材编委会讨论制定的“数据结构”课程教学大纲编写的教材。本书系统地介绍各种类型的数据结构和检索排序的方法。对每一种数据结构从其逻辑结构、抽象运算、存储结构以及在相应存储结构的运算实现都进行了详细的描述，并尽可能地列举各种数据结构的不同应用实例。对每一个算法都用 C 语言描述其实现，并对算法的性能给出定量或定性的分析比较。最后一章讨论了文件的各种组织方法。

本书作者是有着二十多年“数据结构”课程教学与实践经验的教师，全书在教材内容选取上突出应用，内容组织上循序渐进，内容叙述上通俗易懂，课程目标设定上学以致用。全书内容丰富、概念清楚、叙述严谨、可读性强。每章配有丰富的例题、习题和上机实验题，并有可与之配套使用的《数据结构习题解析与上机指导》，便于教师组织教学和学生自学。

本书以知识单元为基本构件，便于拆卸和重组，可以满足不同院校的不同培养层次的教学需求，也可作为从事计算机科学与技术工作的科技人员的参考用书。

本书为授课教师免费提供电子教案，此教案用 PowerPoint 制作，可以任意修改。需要者可从中国水利水电出版社网站免费下载，网址为 http://www.waterpub.com.cn/softdown/。

图书在版编目（CIP）数据

数据结构：用 C 语言描述 / 宁正元，易金聪编著．2 版．—北京：中国水利水电出版社，2006

（21 世纪高等院校计算机系列教材）

ISBN 7-5084-2944-3

Ⅰ．数… Ⅱ．①宁…②易… Ⅲ．①数据结构—高等学校—教材②C 语言—程序设计—高等学校—教材 Ⅳ．①TP311.12②TP312

中国版本图书馆 CIP 数据核字（2006）第 035799 号

书　　名	数据结构——用 C 语言描述（第二版）
作　　者	宁正元　易金聪　编著
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044）
	网址：www.waterpub.com.cn
	E-mail：mchannel@263.net（万水）
	sales@waterpub.com.cn
	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京蓝空印刷厂
规　　格	787mm×1092mm　16 开本　16 印张　384 千字
版　　次	2000 年 6 月第 1 版
	2006 年 5 月第 2 版　2006 年 5 月第 8 次印刷
印　　数	31001—35000 册
定　　价	24.00 元

第二版前言

《数据结构（用C语言描述）》一书自2000年出版以来已经使用了6年，受中国水利水电出版社的委托，结合多年来本课程的教学实践，我们对原书进行了修订。修订的基本原则仍然是：①着重阐明常用数据结构内在的逻辑关系，讨论它们在机内的存储表示，并结合实例说明它们在各种运算操作时的动态性质及执行算法。②在内容的叙述上力求做到符合通俗易懂的原则：在算法的描述上力求结构清晰、描述正确、易读易理解，并对每一个算法都做了大量的注释；在内容的叙述上力求做到由浅入深和深入浅出，用语大众化，增强可读性。

关于第二版《数据结构（用C语言描述）》的修订和编写工作，有以下几点说明：

1．本版继承了第一版的编写思想。继续强调《数据结构》课程的教学要求是使学生学会分析研究计算机加工的数据对象的特征，能够在研发应用实践中适当地选择数据结构和相应的有关算法，初步掌握算法的时间与空间性能分析技巧，得到复杂程序设计的训练。

2．全书在结构上尽量保持与第一版一致，以维持教材的连续性。根据教学经验，增删了一些章节，如在第三章增加了I/O缓冲区管理、优先队列、双端队列；第四章增加了串的堆式动态存储及运算实现、串的块链存储表、第八章增加了B+树等内容，相应的，也删除了一些章节内容，对每一章中的内容如算法部分也进行了修改。

3．数据结构的理论核心是数据结构内在的逻辑关系以及它们在机内的存储表示。本书结合典型应用问题，并给出了算法设计与分析，有利于学生综合分析能力的培养与科学方法论的掌握。在内容组织上，全书以知识单元为基本构件、各知识单元内容相对独立，具有可拆卸性与可重组性，便于不同院校的不同培养层次按需要组织教学。在实例列举上力求恰当，并附有相应的练习题、上机实验题和配套的《数据结构习题解析与上机实验指导》。

本教材的参考学时数为80学时。建议讲授学时为60学时，上机实验学时为20学时。由于书中包含的内容较多而教学时数有限，可根据情况作适当调整。

本书由宁正元、易金聪共同改编，宁正元执笔第一、二、五、七、九章，易金聪执笔第三、四、六、八、十章。王秀丽和张秀萍承担了全书的文字录入、排版、校对工作和部分算法的上机调试，林大辉也提供了一些帮助。陈国龙教授和康宝生教授对本书的编写提供了许多有价值的参考意见；中国水利水电出版社对本书的出版给予了极大的鼓励和支持，作者在此一并表示真诚的感谢。另外，对于本书第一版的原作者朱穗颖、邓桂英、张健、舒坚也表示深深的谢意。

由于时间仓促和作者水平所限，本书一定还存在不少问题，敬请同行专家和广大读者批评指正，我们将不胜感激。

作者

2006年5月

第一版前言

“数据结构”是计算机科学与技术专业教学计划中的一门核心课程，是计算机科学与技术专业和其他有志从事计算机与技术工作的人员的一门重要的专业基础课程。计算机科学与技术学科各领域都要用到各种数据结构，而且要从事计算机科学与技术工作，尤其是计算机应用领域的开发研制工作，必须具备良好的数据结构基础。如语言编译要使用栈、散列表、语法树等；操作系统中要使用队、存储管理表、目录树等；数据库系统要使用线性表、链表、索引树等；人工智能中要使用广义表、检索树、有向图等；软件工程、面向对象的程序设计、计算机图形学、多媒体技术、计算机辅助设计等都要用到许多数据结构。

“数据结构”课程的教学要求是，学会分析研究计算机加工的数据对象的特性，以便在实际应用中选择适当的数据结构、存储结构和相应的算法，初步掌握算法的时间与空间性能分析技巧，得到复杂程序设计的训练。本书旨在阐明常用数据结构内在的逻辑关系，讨论它们在机内的存储表示，并结合实例说明它们在各种运算操作时的动态性质及执行算法。这样，不仅为读者学习后续课程提供了必要的知识准备，而且更重要的是进一步提高了软件设计与编程的能力和水平。

本书内容共分十章。第一章引出数据结构与算法的一些基本概念与术语，并对算法描述及算法分析作了简单说明；第二章到第五章由浅入深地介绍了线性结构中的线性表、栈、队、串、数组及广义表的基本定义、基本算法和基本应用；第六章、第七章介绍了非线性结构的树、二叉树和图；第八章、第九章介绍了在实际应用中使用非常广泛的检索和排序的基本算法，并进行了简单的时间、空间性能分析；第十章讨论了文件及其组织。全书以知识单元为基本构件，各知识单元内容相对独立，具有可拆卸性与可重组性，便于不同院校的不同培养层次按需要组织教学。全书侧重应用性，力求内容与应用实例相结合，有利于学生加深对内容的理解和掌握。

为了便于自学，本书在文字描述上力求通俗易懂；在算法描述上力求结构清楚，正确易懂；在实例列举上力求恰当，并附有相应的练习题和上机实习题。作为高校教材，建议讲授学时为 60 学时，上机实习学时为 20 学时。应该鼓励学生分析和改进算法，提出新的算法，特别是实际编程实现各种算法。强调多做习题和上机实习题的重要性。

本书由华东地区多所高校在第一线讲授数据结构课程的具有丰富教学经验的教师共同编写而成的。本书的第一、二、五、八、九章由宁正元执笔，第三章由张健执笔，第四章由舒坚执笔，第六章由邓桂英执笔，第七、十章由朱穗颖执笔，附录由易金聪整理。全书由宁正元主编、统稿、修改、定稿，西北大学康宝生博士后主审。由于时间仓促和作者水平所限，本书一定还存在不少问题，敬请广大读者批评指正。感谢华东地区计算机教材编委会和中国水利水电出版社、上海交通大学出版社、东南大学出版社对本书的出版给予的支持和鼓励；感谢各有关高校给予作者的大力支持。

作者

2000 年 2 月

目　　录

第 1 章　绪论

随着现代电子计算机技术的发展，计算机的应用领域从最初简单的科学计算逐步发展到人类活动的各个领域；计算机处理的对象从最初简单的数值或字符到现在具有不同复杂结构的各种数据，这就给程序设计带来一些新的问题。为了设计出一个好的算法或程序，除了要掌握所用的计算机语言外，还要研究各种数据的特性和数据之间存在的关系，这就要求加强对数据结构这门课程的学习，以便更有效地使用计算机。

本章将介绍数据结构研究的对象、基本概念和术语、算法的概念及其描述方法（C 语言描述）、数据类型以及抽象数据类型，并概述数据结构的发展概况及其在计算机科学中的地位。

1.1　什么是数据结构

一般来说，用计算机解决一个实际问题时，是先对具体问题抽象，建立起实际问题的求解模型，设计出相应的算法，然后编写程序并上机调试，最终解决实际问题。当人们用计算机处理的是数值计算问题时，所用的求解模型可以用数学方程描述，所涉及的运算对象一般是整型、实型或逻辑型等一些简单类型的数据。此时，程序设计者的主要精力集中于程序设计的技巧上，而对数据的组织和存储不是那么关心。随着计算机应用领域的不断扩大，计算机处理的对象更多的是非数值计算问题，如图书资料查询、电话号码自动管理、交通道路规划、博弈游戏等问题，它们的数学模型无法用数学方程进行描述，此时就必须建立相应的数据结构进行描述，分析问题中所用到的数据是如何组织的，研究数据之间的关系如何，进而为解决这些问题设计出合适的数据结构。

例 1　职工信息管理

一个单位要对所有职工的基本情况进行管理，必须经常了解职工的各种信息，包括经常查询职工的编号、姓名、性别以及月收入等情况，并对整个单位的情况作统计、汇总等工作。如何利用计算机来辅助管理这些信息呢？显然，应先考虑如何对整个单位每个职工的信息进行有效的组织，并考虑如何将这些信息存入计算机中，利用计算机进行有效的处理。

一般可以建立一个花名册对每位职工的信息进行编排，将每位职工的信息组织成如图 1.1 所示的花名册。花名册中每个职工的信息可由编号、姓名、性别、年龄、月收入等项目组成，占表的一行，整个花名册就构成了一个数据结构，或者称整个花名册就是一个数据结构，而把表中的每一行看作一个结点或一个元素、一个记录，它由编号、姓名、性别、年龄、月收入等项目组成，从而整张表就构成了一个关于职工花名册的数学模型，此时的花名册就是一张所谓的线性表，计算机就可以按某个特定要求对其进行操作。表中的结点和结点之间是一种简单的线性关系，表中有且仅有一个结点为表头（第一个结点），有且仅有一个结点为表尾（最后一个结点）；表中任一结点，与它相邻且在它前面的结点最多只有一个；同样，与它相邻且在它后面的结点最多只有一个，这就是上述花名册表的逻辑结构。当考虑用计算机对上述花名册表中的数据进行运算或处理时，就要考虑那些结点如何在计算机中进行存储表示，即存储结构的

问题。对于这样的花名册表，有可能因一些职工调入或调离，必须增加新的结点或将相应结点从表中删除掉。就必须考虑如何进行结点的插入、删除、修改、检索或查找，这就涉及到数据的运算问题。只有弄清这些问题后，才能有效地使用花名册表这个数据结构，有效地解决该单位职工基本信息的计算机辅助管理的问题。

编号	姓名	性别	年龄	月收入
1	李　泉	男	51	980
2	王　怡	女	47	945
3	张　三	男	35	870
4	马小丁	男	27	840
…	…	….	…	…

图 1.1　职工花名册表

例 2　旅游交通网络图

假如要建立一个旅游交通网络咨询系统，用来回答游客提出的各种问题，就必须研究所涉及的各个城市的情况，对这些城市所代表的信息进行分析、组织。此时就可采用一种称之为图的结构来表示实际的交通网络，如图 1.2 所示，图中顶点表示城市，边表示城市间的交通联系，此时，这幅交通网络图就表示一个数据结构。在这个数据结构中，结点之间的关系可以是任意的，图中任意两个结点之间都可以相关。

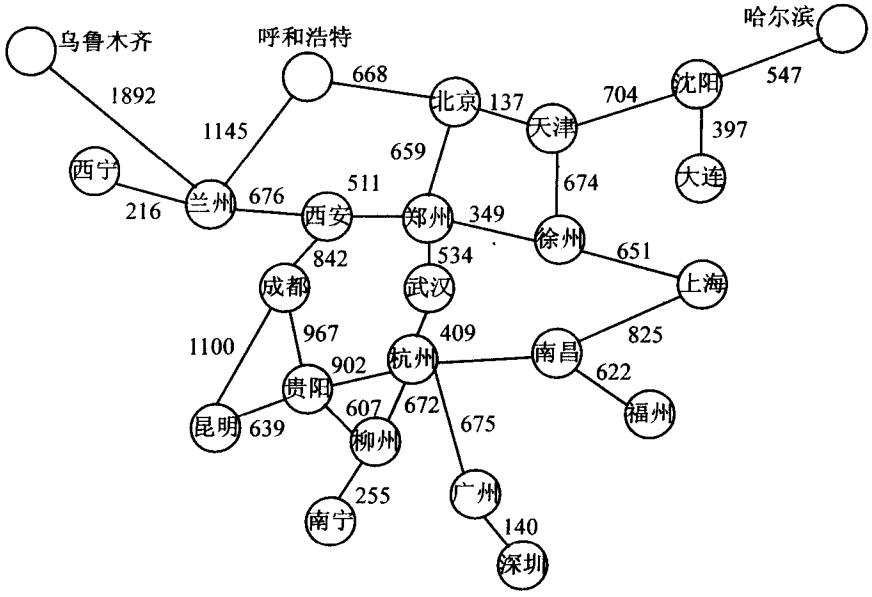

图 1.2　一幅表示交通网的例图

构造出这幅交通图后，若要利用计算机进行处理，还得先讨论如何将这幅图存放入计算机中，这就必然涉及到所谓图的存储结构问题，即将图中结点本身的值以及每个结点之间的可能的关系都存入计算机，然后再利用计算机进行操作处理，这就是所谓的图的运算。如一位旅

客要从 A 城市到 B 城市，他希望选择一条旅途中转次数最少的路线，假设图中每一站都需要换车，这个问题反映到图上就是要找一条从顶点 A 到顶点 B 包含边的数目最少的路径，也就是所谓求解图的最短路径的运算问题；还有一些诸如关键路径等问题都可认为是对图这种数据结构所作的运算。

从以上两个例子可见，要描述诸如此类的非数值计算问题的数学模型，仅仅依靠数学方程是无法满足的，它涉及到一些诸如表、图还有树之类的数据结构。数据结构课程实际上是一门研究非数值计算问题的程序设计中计算机的操作对象以及它们之间的关系和运算操作等的一门学科。它和高级程序设计语言课程不同，高级程序设计语言课程所讨论的是简单程序设计问题，而数据结构课程所讨论的是在数据的结构组织基础上的复杂程序设计问题。

《数据结构》作为一门独立的课程在国外是从 1968 年才开始设立的，1968 年在美国一些大学的计算机系的教学计划中，把《数据结构》规定为一门课程，但对课程的范围没有作明确规定。当时，数据结构几乎和图论，特别是和表、树的理论等视为同义词。随后，数据结构这个概念被扩充到包括网络、集合代数论、格、关系等方面，从而变成了现在称之为《离散结构》的内容。然而，由于数据必须在计算机中进行处理，因此，不仅要考虑数据本身的性质，而且还必须考虑数据的存储结构，这就进一步扩大了数据结构的内容。

从 20 世纪 60 年代末到 70 年代初，出现了大型程序，软件也相对独立，结构化程序设计成为程序设计方法学的主要内容，人们就越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的结构，加上设计一种好的算法，正如瑞士著名的计算机科学家 N.Wirth 所提出的著名的公式“程序=算法+数据结构”所阐述的那样。从 20 世纪 70 年代中期到 80 年代初，各种版本的数据结构著作就相继出现。

目前在我国，《数据结构》已不仅仅是计算机科学技术专业的教学计划中的核心课程之一，也是其他非计算机专业的主要选修课程之一。

《数据结构》在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及到计算机硬件（特别是编码理论、存储装置和存取方法等）的研究范围，而且和计算机软件的研究有着更密切的关系，无论是编译程序还是操作系统，都涉及到数据元素在存储器中的分配问题。在研究信息检索时也必须考虑如何组织数据，以便查找和存取数据元素更为方便。因此，可以认为数据结构是介于数学、计算机硬件和计算机软件三者之间的一门核心课程。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础。

随着计算机科学技术的发展，计算机应用的普及，仅仅掌握计算机语言和程序设计的技巧和方法，而缺乏有关数据结构的知识，几乎很难应付复杂的课题，更不能有效地利用计算机。

1.2　基本概念和术语

数据（Data）是指信息的载体，是对客观事物的符号表示，它能够被计算机识别、存储和加工处理，即数据是对有效地输入到计算机中并能被计算机程序处理的符号的总称。例如，解代数方程程序中所用到的整数和实数，文本编辑中所用到的函数和字符串等，都是程序加工处理的对象，客观事物中的声音、图形、图像，经过某些处理也可使计算机识别、存储和处理，所以它们也可属于数据的范畴。

数据元素（Data Element）是数据的基本单位，有时也称之为结点、元素、顶点、记录等。有时一个数据元素可由若干个数据项（Data Item）组成，如上面职工花名册中的每个人的信息就是一个数据元素，而每个人的信息包括编号、姓名、性别、年龄、月收入等，这每一个项目就属于数据项。数据项是数据的不可分割的最小单位。

数据对象（Data Object）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，+1，-1，+2，-2，...}，字母字符数据对象是集合C={'A'，'B'，'C'...'Z'}。

数据类型（Data Type）是对在计算机中表示的同一数据对象及其在该数据对象上的一组操作的总称。如整数，在计算机中它是[minint...maxint]上的整数，在这个集合上可以进行加、减、乘、整除、求模等一些基本操作（minint、maxint称作最小、最大整数，在不同的计算机中它的值不同）。

数据类型有时还分为原子数据类型和结构数据类型。原子数据类型是由计算机语言系统提供的一些简单类型，其值不可分解，如整型、实型、字符型；而结构数据类型是借用计算机语言中类型定义的有关语法机制来定义的，或可看作是由原子数据类型用某种方式（可以简单也可以复杂）组合而成的，因而其值是可以分解的，其成分可以是原子的，也可以是结构的，不同的组合方式形成不同的结构类型，例如C语言中的结构体、数组、文件、共同体结构等。

数据结构（Data Structure）指的是数据及数据之间的相互关系。由1.1节例子可以发现在任何问题中，数据元素都不是孤立的，它们之间具有某种关系，这种数据元素相互之间的关系有时也称为结构。一般地，数据结构包括以下三个方面的内容。

（1）数据元素之间的逻辑关系，有时也称为数据的逻辑结构（Logical Structure）；

（2）数据元素及其关系在计算机内存中的表示（又称为映像），称之为数据的物理结构，又称为数据的存储结构，它包括数据元素的表示和数据元素之间关系的表示；

（3）数据的运算及实现，即对数据元素可以施加的操作以及这些操作在相应的存储结构上的实现。

对于数据结构三个部分的理解，我们必须知道数据的逻辑结构是从逻辑关系上描述数据，它与数据的存储无关，是独自于计算机的。因此，数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。如上例中的职工花名册表，它仅表示的是每个职工信息（数据元素）之间的关系而已，只是在逻辑关系上说明每个职工信息之间的相互关系。根据数据元素之间的不同特性，通常有下列四类基本逻辑结构。

（1）线性结构。结构中的数据元素存在一对一的关系，即所谓的线性关系，如1.1节例1中的职工花名册就可看作是一个典型的线性结构，所以又称为线性表。这个表中的数据元素之间的逻辑关系是：对表中任何一个结点，与其相邻且在它之前的结点（有时称作直接前趋）最多只有一个，与表中任一结点相邻且在其后的结点（又称为直接后继）也最多只有一个，且在这个表中只有第一个结点没有前趋，称为开始结点，同样也只有最后一个结点无后继，称为终端结点，其他结点之间都是相邻的，且一个跟着一个。也就是说这个表的数据元素之间的关系具有线性结构的逻辑特征，有且仅有一个开始结点和一个终端结点，并且所有结点最多只有一个直接前趋和一个直接后继。

（2）树形结构。该结构中的数据元素之间存在一对多的关系。树形结构实际上是结点之间有分支并具有层次关系的结构，它非常类似于自然界中的树。例如，平常所说的家谱、行政组织机构等都可用树形象地表示，即可将其抽象为成树形结构来处理，在这种结构中，每个结

点允许有多个直接后继。

（3）图或网状结构。该结构中的数据元素之间的关系存在多对多的关系，如 1.1 节例 2 中所提到的交通网络图就是一个图状结构。对图结构而言，其结点（图中常称为顶点）的前趋和后继个数都不加限制，结点之间的关系则是任意的，图中任意两个结点之间都可以相关。

（4）集合。结构中的数据元素之间除了“同属于一个集合”的相互关系外，别无其他关系。

有时将树形结构、集合和图或网状结构称为非线性结构，此时数据逻辑结构就可分为线性结构和非线性结构两类。

数据的存储结构则是指逻辑结构用计算机语言的实现，也就是数据元素及其关系如何存放入计算机内存的问题。就如 1.1 节例 1 中，要在机内存储职工花名册表，就必须考虑如何存储每个职工的信息本身，同时还必须考虑如何将每个职工信息之间的相互联系在机内体现出来，也就是必须存储它们之间的相互关系。一般地，数据的存储结构可按以下四种基本存储方法而得到。

（1）顺序存储方法。该方法是把逻辑上相邻的结点存储在物理位置上相邻的存储单元中，结点间的逻辑关系由存储单元的相邻关系来体现。由此得到的存储表示称为顺序存储结构（Sequential Storage Structure），通常可借助计算机语言的数组来描述。该方法可用于线性结构的存储，如 1.1 节例 1 中的职工花名册就可利用一个一维数组来存储，对于一些非线性结构也可以通过某种线性化处理后实现顺序存储。

（2）链状存储方式。该方法不要求逻辑上相邻的结点在物理位置上也相邻，数据元素可以存储在任意位置。显然，为了实现数据元素之间逻辑关系的存储，就必须通过一些附加的手段来存储这种相互关系。由此得到的存储表示称为链状存储结构（Linked Storage Structure）。一般可以用指针（Pointer）来实现，通常可借助计算机程序语言的指针类型或者游标来描述。

（3）索引存储方法。该方法通常是在存储结点信息的同时，建立附加的索引表，索引表中有多个索引项，索引项的一般形式可为（关键字，地址），其中关键字是指可标识一个结点的数据项。索引表一般有稠密索引（Dense Index）和稀疏索引（Sparse Index）两种，如果每个结点在索引表中都有一个索引项，该索引表称为稠密索引，稠密索引中索引项的地址可指出结点所在的存储位置；若一组结点在索引表中只对应一个索引项，则该索引表称为稀疏索引，稀疏索引中的索引项可指示一组结点的起始存储位置。

（4）散列存储方法。该方法依据结点的关键字直接计算出该结点的存储地址，然后将结点按某种方式存入该地址。

一般而言，以上四种基本的存储方法，既可单独使用，也可以组合起来对数据结构进行存储。同一种逻辑结构采用不同的存储方法，可得到不同的存储结构。选择何种存储结构来存储表示相应的逻辑结构就要视具体问题的要求而定，也可依据运算是否方便和算法的时间效率和空间要求而定。显然，它是依赖于所用的计算机语言的，一般可借助高级程序语言来描述数据的存储结构。

数据的运算是指定义在数据的逻辑结构上的一组操作的集合。每一种逻辑结构都允许有某些运算。例如，一些常用的如检索（查找）、插入、删除、定位、修改、排序等。要注意的是，这些运算实际上是在逻辑结构上对抽象数据所施加的一系列抽象的操作，对于这些操作，只要知道这些操作可以“做什么”，而无须考虑它们是“如何实现”，只有对某种数据结构选定了存储结构之后，此时才考虑如何地将这些运算具体地实现。也就是说，运算的实现依赖于所

选取的存储结构，是依赖于不同的计算机程序设计语言的。本书所讨论的运算，均以C语言描述的算法来实现。

1.3 数据类型和抽象数据类型

数据类型是具有相同性质的计算机数据的集合及在这个数据集合上的一组运算，是和数据结构密切相关的概念。在用高级程序语言编写的程序中，每个变量、常量和表达式都有一个它所属的数据类型，类型明显或隐含地规定了在程序执行期间变量或表达式所有可能的取值范围以及在这些数值范围上的所允许进行的操作。例如，整型数，它是[-minint...maxint]区间上的整数，在这个整数集上就可以进行加、减、乘、整除、取模等操作。在高级程序语言中，数据类型按照它的结构是否可以分解，可以分为原子类型和结构类型。原子类型即值不可分解的类型，如C语言中的整型、字符型、实型、指针类型等；而结构类型则指的是其值可由若干成分按某种结构组成的类型，是可以分解的，它所包含的各成分可以是原子类型也可以是结构类型。如C语言中提供的结构体，它相当于其他高级程序语言中的记录，可由许多成员（或称分量）所组成，每一个成员称为结构体中的域，可以是整型或字符型，也可以是结构体或数组等。

在分析一些复杂的情况或处理复杂的事务时，一般采用抽象思维的方法，即舍去复杂系统中非本质的成分，只将其中一些本质的可以反映系统重要特性的东西归纳提炼出来，设计出系统的模型，然后深入研究这些模型，得出一般规律性的东西，然后解决实际问题。在复杂的软件系统设计中同样可以应用抽象思维的方法，对软件系统中所包括的数据和操作进行抽象分析，从而更有效地进行软件系统的设计。将数据抽象和运算抽象紧密地结合在一起就可得到所谓的抽象数据类型。

抽象数据类型（Abstract Data Type，简称ADT）是指一个数学模型及定义在该数学模型上的一组操作。抽象数据模型的定义取决于它的一组逻辑特性，与其在计算机内部如何表示和实现无关。换句话说，不论它的内部结构如何变化，只要它的数学特性不变，都不影响它的外部使用。

抽象数据类型是算法设计和程序设计中的重要概念。严格地说，它是算法的一个数据模型连同定义在该模型上的一组运算，从而明确地把数据模型与作用在该模型上的运算紧密地联系起来。数据模型的运算依赖于数据模型的具体表示，因为数据模型的运算是以数据模型中的数据变量作为运算对象或作为运算结果的，或二者兼而有之；另一方面，数据模型的具体表示以及在数据模型上运算的具体实现一旦确定，运算的效率也就随之确定。于是就有了这样一个问题，如何选择数据模型的具体表示使该模型的各种运算的效率尽可能地提高？很明显，对于不同的运算组，为使组中所有运算的效率都尽可能高，其相应的数据模型具体表示的选择将是不同的。在这个意义下，数据模型的具体表示又反过来依赖于数据模型上定义的那些运算，数据模型与定义在该模型上的运算之间存在着的这种密不可分的联系，是抽象数据模型的概念产生的背景与依据。

应该指出，抽象数据类型的概念并不是全新的概念，它实际上是我们熟悉的基本数据类型概念的引申和发展。由计算机高级程序语言知识可知，基本数据类型已隐含着数据模型和定义在该模型上的运算的统一，只是当时还没有形成抽象数据类型的概念罢了。如整型就是整数

数据模型和加、减、乘、除、取摸等几种运算的统一体。每一种基本数据类型都连带着一组基本运算，只是由于这些基本数据类型中的数据模型的具体表示和基本运算的具体实现都很规范，都可以通过内置而隐藏起来，使人们看不到它们的封装。我们只是习惯于在实际设计中用基本数据类型名和相关的运算名，没有意识到抽象数据类型的概念实际上已经隐含在基本数据类型的概念之中。

要定义抽象数据类型，可以依据抽象数据类型的概念，首先约定抽象数据类型的名字，同时，约定在该类型上定义的一组运算的各个运算的名字，明确各个运算分别要有多少个参数，这些参数的含义和顺序以及运算的功能。一旦定义清楚，就可十分简便地引用抽象数据类型。例如，可以定义线性表的抽象数据类型如下：其中数据关系可以是在对数据对象分析后建立起来的一种数据模型。

ADT LIST
{数据对象：D={ a|ai∈elemset，i=1，2，…，n；n≥0 }
　　/*定义线性表，它是具有 n 个数据元素的有限序列*/
数据关系：R={< ai-1，ai > ai-1，ai∈D　i=1，2，…，n　}
　　/*说明线性表数据元素之间的相互关系*/
基本运算：
SETNULL(L)
初始条件：线性表 L 存在；
运算结果：将线性表 L 置成空表；
LENGTH(L)
运算结果：返回 L 线性表的数据元素个数，即表长；
GET(L,i)
初始条件：线性表 L 存在，且 1≤i≤LENGTH(L)；
运算结果：返回表 L 的第 i 个结点；
LOCATE(L,x)
初始条件：线性表 L 中存在一个值为 x 的结点；
运算结果：返回该结点的位置。若表 L 中存在多个值为 x 的结点，则是 L 中首次找到的结点位置，当表中不存在值为 x 的结点时，将给出一个特殊值，如：0 表示值为 x 的结点不存在。
INSERT(L,x,n)
初始条件：线性表存在，且 1≤i≤LENGTH(L)+1；
运算结果：在线性表 L 的第 i 个位置插入一个值为 x 的新结点，使原编号为 i，i+1，…，n 的结点变为编号为 i+1，i+2，…，n+1 的结点；
DELETE(L,i)
初始条件：线性表已存在且不空，1≤i≤LENGTH(L)；
运算结果：删除线性表 L 的第 i 个结点，使得原编号 i+1，i+2，...，n 的结点变成编号为 i，i+1，...，n-1 的结点。
} ADT　LIST;

由上面所介绍的线性表抽象数据类型的定义可见，抽象数据类型实际上是数据类型的进