HZ BOOKS
华章教育

高等院校计算机教材系列

算法与数据结构
C语言版
第2版

陈守孔 孟佳娜 武秀川 等编著

为教师配有电子教案

机械工业出版社
China Machine Press

高等院校计算机教材系列

算法与数据结构
C语言版
第2版

陈守孔 孟佳娜 武秀川
胡潇琨 陈 卓 张东娜 编著

机械工业出版社
China Machine Press

本书以通俗的语言，按照由易到难的原则，详细介绍了各种数据结构的基本概念、逻辑特性和物理特性，对各种结构定义了相应的抽象数据类型（ADT）。在各章末尾，还给出了算法设计举例和习题。

本书可作为高等院校计算机及相关专业的教材，同时可供计算机科学及工程技术人员参考。

图书在版编目（CIP）数据

算法与数据结构（C语言版）第2版 / 陈守孔等编著. —北京：机械工业出版社，2008.1
（高等院校计算机教材系列）
ISBN 978-7-111-14620-9

Ⅰ.算…　Ⅱ.陈…　Ⅲ.①数据结构-高等学校-教学参考资料　②电子计算机-算法设计-高等学校-教材　Ⅳ.TP311.12

中国版本图书馆CIP数据核字（2004）第057884号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：王春华
三河市明辉印装有限公司印刷·新华书店北京发行所发行
2008年6月第2版第2次印刷
184mm×260mm · 17印张
标准书号：ISBN 978-7-111-14620-9
定价：28.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

第1版前言

“数据结构”是20世纪70年代开始设立的计算机科学与技术专业的基础课，现在是其重要的核心课程。随着计算机软、硬件技术的发展，计算机在各个学科和领域得到广泛应用，其当今应用的一个主要方面就是数据处理，如情报检索、数据采集、企业管理、决策分析和人工智能等。而应用到这些领域时所面临的首要问题就是对于信息量大、种类繁多、结构复杂的数据和数据关系的处理，由此必须设计高级的数据结构和数据组织，以便有效地实现数据采集、数据组织、数据存储、数据传输和数据处理。数据结构主要研究数据的逻辑结构、数据在计算机中的物理结构，以及处理不同结构数据的算法。我们研究数据结构的目的是为了学会编写更高效的程序，而高效、简洁的程序取决于数据结构和算法的设计。

现在，有关算法与数据结构的教材，一部分侧重于算法的基本思想和步骤，用伪代码描述；一部分用程序语言描述，侧重于算法的程序实现。前者侧重于对数据结构原理的阐述，不利于学生掌握算法的程序实现以及对算法的分析、比较。后者增加了对数据元素的描述，从而使得程序更难理解。

本教材对数据结构和算法用类C语言描述，数据结构和算法在设计本质上是属于底层的，在本书中并不提供可直接上机运行的源代码，而是提供一种类C语言伪代码来描述算法的基本思想和基本步骤。此外，用类C语言描述的算法容易转换为C语言程序。

本书系统全面地介绍了各种传统的数据结构，对每种数据结构及相关算法都进行了时间和空间效率的分析，强调算法与数据结构的密不可分性，引进抽象数据类型概念，将数据类型与其上的操作封装为一体，为面向对象的程序设计方法奠定了坚实的基础。此外，针对不同的抽象数据类型讨论不同的存储方法，并且研究不同存储方法的可能算法，从而体现了计算机学科方法论的理论、抽象和设计三个过程。

全书分为11章：第1章是概论，解释了从问题到程序的求解过程，以及在这一过程中抽象数据类型的表述和作用，介绍了程序的运行步骤及复杂度；第2章介绍了有关线性表的概念及其基本操作，该章是后续章节的基础；第3章在第2章的基础上讨论了操作受限的线性表——栈与队列的特点，并列举了几个应用示例；第4章简要给出了非数值处理——串的处理方法；第5章主要是关于程序设计中的一种常用数据类型——数组在计算机内部的表示和实现，同时介绍了广义表的概念；第6章描述了非线性复杂数据结构——树，它是解决决策和博弈等问题的基本结构；第7章是关于图的内容，包括有向图和无向图。图是一种比树更复杂的数据结构；第8章涉及存储管理中应用的基本方法；第9章以集合作为数据模型，讨论了查找的方法和技术；第10章介绍了一些常用的排序算法，包括内部排序和外部排序；第11章简要介绍了文件。

本书由教学一线的教授亲自主笔，根据作者多年的教学实践经验，针对算法与数据结构这门课程的特点撰写而成，目的是培养学生合理组织数据和进行优秀的算法设计的能力。本书尽量合理地安排内容顺序，教师可以根据需要自由地重新组织内容。

本教材可作为高等院校计算机科学与技术专业的教材、参考书和考研辅导，同时也可供计算机工程技术人员参考。对于计算机科学与技术专业，可讲授64学时，对于其他专业，去

掉带星号的章节，可讲授48学时。

本书的第1、2、3章由周世平同志编写，第4、10章由胡潇琨同志编写，第5、9章由孟佳娜同志编写，第6、8、11章由谭征同志编写，第7章由范策同志编写。全书由范策同志统稿，陈守孔同志编写了各章最后一节的算法设计并审阅了全书，孟佳娜同志编写了本书的电子教案。

由于作者水平所限，加上计算机科学技术的发展十分迅速，书中难免有不妥和挂一漏万之处，恳请广大读者赐教。

编　者

2004年4月于烟台大学

第2版前言

“数据结构”是计算机专业的一门专业基础课，也是重要的核心课程之一。随着计算机技术的飞速发展，计算机在各个学科和领域得到了广泛应用，而应用过程中所面临的首要问题就是对于信息量大、种类繁多、结构复杂的数据和数据关系的处理，由此必须设计出好的数据结构，以便有效地实现数据存储、数据传输和数据处理等操作。

数据结构课程主要培养以下几个方面的知识和能力：1）掌握并能根据实际问题灵活应用基本数据结构的抽象数据类型、存储方法和主要算法；2）掌握基本的算法设计和分析技术；3）掌握并能应用常用的排序、查找方法；4）具备一定的调试算法和程序、项目测试的能力。显然，合理地组织数据、有效地表示数据和正确地处理数据，这三者是提高程序设计质量的关键因素。

本书的特点是注意基本概念的引入和阐述，注重算法设计的分析方法，强调实践环节的重要性。本书的第1版已由机械工业出版社于2004年出版，经过几年的教学使用和实践，同时也结合该课程的发展，本书第2版对书中的部分内容进行了修改。本教材沿用了第1版教材的算法描述方式，即采用类C语言的描述方法。

考虑到初学者对算法设计问题普遍感到比较困难，思路不明确，本书每一章都设有算法设计举例，旨在提高初学者的算法分析和设计能力。每章之后附有习题，以便读者进一步练习并检验学习效果。鉴于本课程是实践性较强的一门课程，本教材在附录中设立了实验题目，以便使读者在掌握理论内容的同时，通过实践环节加深理解。

本书由长期从事数据结构课程教学的教师编写，总结了作者多年的教学实践经验。本书尽量合理地安排内容顺序，教师可以根据内容需要自由地重新组织内容。

本书可作为高等院校计算机及相关专业的教材、参考书和考研辅导，同时也可供计算机科学及工程技术人员参考。对于计算机科学与技术专业，可讲授64学时，对于其他专业，可去掉带星号的章节，讲授48学时。

由于工作的变动，本书第2版重新组织了编写人员，对于第1版的作者表示衷心感谢。

本书第1章由武秀川同志编写，第2、3章由陈守孔同志编写，第4、10章由胡潇琨同志编写，第5、9章由孟佳娜同志编写，第6、8章由陈卓同志编写，第7、11章由张东娜同志编写。孟佳娜同志校阅了各章，并编写了实验题目，陈守孔同志编写了各章习题并对全书统稿定稿。陈卓、孟佳娜和陈守孔同志提供了本书的电子教案。

本书的出版得到机械工业出版社温莉芳女士和王春华编辑的大力支持，在此深表感谢。

由于作者水平有限，加上计算机科学技术的发展十分迅速，书中难免有不妥和挂一漏万之处，恳请广大读者赐教。陈守孔的电子信箱是：skcnmu@163.com，孟佳娜的电子信箱是：ytumengjn@163.com，武秀川的电子信箱是：wxc225@126.com。

编　者

2007年10月

目 录

第1章 概 论

算法与数据结构是计算机科学与技术、软件开发与应用、信息管理、电子商务、网络安全等相关专业的一门专业基础课，它不仅是计算机学科的理论基础之一，也是计算机系统软件和应用软件开发者的必备基础，读者无论是从事计算机行业，还是希望在计算机方面继续深造，该课程的学习都是必须的。

算法与数据结构理论发展至今，已成为一门比较成熟的课程。它的应用范围已经渗透到编译系统、操作系统、数据库、人工智能、信息科学、企业管理、系统工程、应用数学、计算机辅助设计及其他信息管理的应用中。

算法与数据结构专门研究从解决非数值计算的现实问题中抽象出来的数据在计算机中如何表示、快速存取和处理的方法。这里所说的数据是广义的概念，它不仅包括数值数据、字符数据、逻辑数据等简单数据，而且还包括带有一定结构的各种复杂的数据，如字符串、记录、向量、矩阵等，也包括各种表格、图形、音频和视频。

当用计算机存储数据时不仅要存储这些数据的值，而且相应地还要存储这些数据之间的相互关系。如何存储数据和这些数据之间的关系就出现了各种不同的存储方法。

1.1 什么是数据结构

计算机的应用可以分为科学计算和生产过程自动控制、管理以及数据处理。一般来说，用计算机解决一个实际问题时，需要经过以下几个步骤：首先从具体问题抽象出一个适当的数学模型，其次选择或设计一个解此数学模型的算法，最后编写程序进行调试、运行，直至得到最终的解答。在此过程中寻求数学模型的实质是分析问题，从中提取操作对象，并找出这些操作对象之间的关系，然后用数学语言加以描述。例如，求解建筑结构工程中的结构静力的分析计算，首先要利用有限元的分析方法得到一个线性代数方程组的数学模型；利用计算机进行全球天气预报需要求解一组球面坐标系下的一般环流模式方程，这个复杂的方程就是天气预报问题的数学模型。这些问题中涉及大量的数值计算，它们的数学模型可以用微分方程、常微分方程、多元函数微分方程、线性微分方程、代数方程、积分方程等表示。但是多数非数值计算问题无法用数学方程进行描述。请看如下3个例子。

例1.1 考生录取信息系统。

若某高校需要在报考该学校的考生中查找某个考生的有关情况，或者想查询报考某个专业的考生的有关情况，或者统计某个分数段的考生的情况，则可以建立相关的数据结构，并且按照某种算法编写相关程序，这样就可以实现计算机自动检索。因此，可以在考生录取信息系统中建立一张按考号顺序排列的考生信息表和分别按姓名、专业、成绩顺序排列的索引表，如图1-1所示。由这四张表构成的文件便是学生信息检索的数学模型，计算机的主要操作有按照某个特定要求（如给定考号）对考生信息文件进行查询。

诸如此类的表结构还有学生学籍管理系统、电话自动查号系统、图书馆的书目管理系统、仓库管理系统和人事档案管理系统等。在这类问题中，计算机处理的对象是各种表，元素之间存在着一种简单的线性关系，施加于对象上的操作有查询、插入和删除等，因此这类问题的数学模型就是各种表格，而插入和删除等操作都是以查找为基础，所以查找算法是解决这

类问题的主要算法。

考号	姓名	性别	报考专业	成绩
2300411	李闽志	男	计算机科学与技术	658
1000472	于惠芳	女	英语	632
1506302	刘　红	女	应用数学	617
2105902	宋大明	男	英语	600
0934785	高大庆	男	计算机科学与技术	601
0600807	何文丽	女	英语	611
0878529	隋文涛	男	应用数学	612
1690834	崔秀海	男	英语	602
1710641	于众群	女	计算机科学与技术	619

a) 考生信息表

崔秀海	8
高大庆	5
何文丽	6
李闽志	1
刘　红	3
宋大明	4
隋文涛	7
于众群	9
于惠芳	2

b) 姓名索引表

计算机科学与技术	1,5,9
英语	2,4,6,8
应用数学	3,7

c) 专业索引表

600~609	4,5,8
610~619	3,6,7,9
620~629	
630以上	1,2

d) 成绩索引表

图1-1　考生信息系统中的数据结构

例1.2　人-机博弈。

计算机之所以能和人博弈是因为已经将对弈的策略输入到了计算机中。由于对弈的过程是在一定的规则下随机进行的，因此，为使计算机能够灵活对弈，就必须把对弈过程中所有可能发生的情况以及相应的对策都加以考虑，并且，一个“好”的棋手在对弈时不仅要看棋盘的状态，还要预测棋局发展的趋势，直至最后结局。所以在对弈问题中，计算机操作的对象是对弈过程中可能出现的称之为格局的棋盘状态。图1-2所示的井字棋对弈树，包括了多个对弈的格局，格局之间的关系是由比赛规则决定的。通常，这个关系是非线性的，因为从一个棋盘格局可以派生出几个格局，而从每一个新的格局又可派生出多个可能的格局。因此，如果将从对弈开始到结束的过程中所有可能出现的格局都表示出来，就可以得到一棵“树”。其“树根”是对弈开始之前的棋盘格局，而所有的叶子就是可能出现的结局，对弈的过程就是从树根沿树杈到每个叶子的过程。

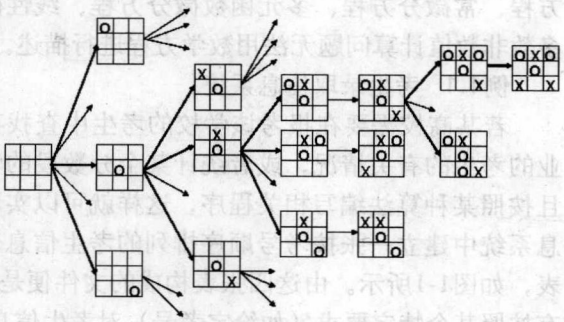


图1-2　井字棋对弈树

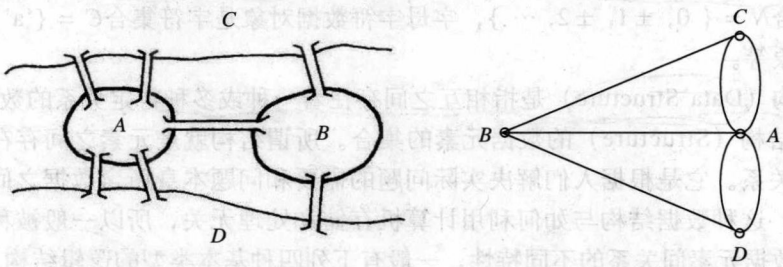
诸如此类的树结构还有家族的族谱、计算机文件系统以及一个单位的组织机构等。在这类问题中，计算机处理的对象是树形结构，元素间的关系是一种层次关系，施加于对象上的操作有查询、插入和删除等，此类问题的数学模型就是如何表示棋盘和棋子，算法就是博弈的规则和策略。

例1.3　哥尼斯堡七桥问题。

七桥问题十分有趣，对于图论这一学科的建立也有很重要的意义，它提供了一个很好地把现实生活模型抽象为图问题的实例。问题的背景如下：在18世纪东普鲁士的哥尼斯堡有许多人热衷于这样一个游戏：由于哥尼斯堡被普莱格尔河分成四块，它们之间通过七座桥互相连接，如图1-3a所示，人们想知道，怎样才能够从某块陆地出发，经过每座桥一次且仅一次最后回到出发点。这一问题一直没有人找到答案。1736大数学家欧拉把两个小岛和南北两岸抽象为四个点A、B、C、D，而把这些桥抽象为连接两个点的一条线，这样，哥尼斯堡七桥问题的图表示如图1-3b所示。欧拉证明，如果存在经过每条边一次且仅一次回到出发点的路径，则充分必要条件是：

1) 图是连通的。

2) 在图中与每个顶点相连的边数必须是偶数。



a) 哥尼斯堡七桥　　b) 哥尼斯堡七桥问题的图表示

图1-3　哥尼斯堡七桥问题

由于哥尼斯堡七桥问题不满足这样的条件，所以它是无解的。欧拉由七桥问题所引发的对图的研究论文是图论的开篇之作，因此欧拉也被称为图论之父。从七桥问题可以看到，要利用图来解决问题，关键的一步是找到现实问题的实体与图的点和边的对应关系。

诸如此类的结构还有城市网络交通图和网络工程图等。这类问题中计算机处理的对象是各种图，元素间的关系是复杂的图形或网状关系，是一种多对多的关系，施加于对象上的操作依然有查询、插入和删除等，此类问题的数学模型就是图或网络，算法是如何求两点之间的距离或最短路径等。

上述三个例子表明，描述这类非数值计算问题的数学模型不再是数学方程式，而是诸如表、树和图之类的数据结构。因此，简单地说，数据结构是一门研究非数值计算程序设计问题中计算机的操作对象以及它们之间的关系和操作的学科，用以描述现实世界实体的数学模型（非数值计算）及其上的操作在计算机中的表示和实现。

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密切的关系，无论是编译程序还是操作系统都涉及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，以使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、计算机硬件和软件三者之间的一门核心课程。

1.2　数据结构的基本概念和术语

在系统地学习算法与数据结构知识之前，先对相关的概念和术语赋予确切的含义。

数据（Data）是信息的载体，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识别和处理的符号的集合。数据是计算机处理的信息的某种特定的符号表示

形式，如数学计算中所用到的整数和实数，文本编辑所用到的字符串等都是数据。随着计算机软、硬件技术的发展，计算机能够处理的对象也在扩大，相应的数据的含义也被拓宽了。文字、图像、图形、声音和视频等非数值数据也都是计算机可以处理的数据。

数据元素（Data Element）是数据中的一个“个体”，是数据的基本单位。在有些情况下，数据元素也称为元素、结点、顶点、记录等。数据元素用于完整地描述一个对象，如一个考生记录、树中棋盘的一个格局（状态）、图中的一个顶点等。

数据项（Data Item）是组成数据元素的、有特定意义的、不可分割的最小单位。如构成一个数据元素的字段（域、属性等）可称之为数据项。又如考生信息表中的学号、姓名、性别、成绩等。

数据元素是数据项的集合。

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。如整数数据对象是集合$N = \{ 0, \pm 1, \pm 2, \cdots \}$，字母字符数据对象是字符集合$C = \{\text{'a'}, \text{'b'}, \cdots, \text{'z'}\}$，学生数据对象等。

数据结构（Data Structure）是指相互之间存在着一种或多种特定关系的数据元素的集合。简言之是带结构（Structure）的数据元素的集合。所谓结构就是元素之间存在的一种或多种特定的约束关系。它是根据人们解决实际问题的需要和问题本身所含数据之间的内在联系而抽象出来的。这种数据结构与如何利用计算机存储和处理无关，所以一般被称为数据的逻辑结构。根据数据元素间关系的不同特性，一般有下列四种基本类型的逻辑结构：

1) 集合结构　集合结构中的数据元素除了仅仅同属于同一个集合外，不存在逻辑关系。

2) 线性结构　线性结构中的数据元素之间存在着一种一对一的关系。这种结构的特征是：若结构是非空集，则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能有一个（直接）前驱和一个（直接）后继。

3) 树形结构　树形结构中的数据元素之间存在着一种一对多的关系。在这种结构中，除了一个特殊的结点（称之为根结点）外，其他所有结点都有且仅有一个前驱结点和零至多个后继结点。

4) 图形结构　图形结构中的数据元素之间存在着一种多对多的关系。在这种结构中，所有结点均可以有多个前驱和多个后继。图形结构也称为网状结构。

数据结构的形式化定义为：数据结构是一个二元组

$$Data\_Structure = (D, R)$$

其中，D是数据元素的有限集合，R是D上关系的有限集合，这里每个关系都是从D到D的关系。在表示每个关系时，用尖括号表示有向关系，如〈a，b〉表示存在结点a到结点b之间的关系；用圆括号表示无向关系，如(a，b)表示既存在结点a到结点b之间的关系，又存在着结点b到结点a之间的关系。设r是一个D到D的关系，$r \in R$，若d，d'$\in D$，且〈d，d'〉$\in r$，则称d'为d的后继结点，d是d'的前驱结点，这时d和d'是相邻的结点（都是相对r而言的）；如果不存在一个d'，使得〈d，d'〉$\in r$，则称d为r的终端结点；如果不存在一个d'，使得〈d'，d〉$\in r$，则称d为r的开始结点；如果d既不是终端结点也不是开始结点，则称d是内部结点。

例1.4　一个12位的十进制数可以用三个4位的十进制数表示：

$$6524，3587，1496 \longrightarrow a_1(6524)，a_2(3587)，a_3(1496)$$

a_1，a_2，a_3之间存在“次序”关系：〈a_1，a_2〉，〈a_2，a_3〉

很显然，

6524，3587，1496 ≠ 3587，6524，1496，也就是说，

$$a_1 \quad a_2 \quad a_3 \neq a_2 \quad a_1 \quad a_3$$

例1.5　一个2行3列的二维数组{a_1，a_2，a_3，a_4，a_5，a_6}

a_1	a_2	a_3
a_4	a_5	a_6

存在着两种关系，分别是行的次序关系：

$$row = \{ \langle a_1, a_2\rangle, \langle a_2, a_3\rangle, \langle a_4, a_5\rangle, \langle a_5, a_6\rangle \}$$

和列的次序关系：

$$col = \{ \langle a_1, a_4\rangle, \langle a_2, a_5\rangle, \langle a_3, a_6\rangle \}$$

显然，数组

a_1 a_3 a_5

a_2 a_4 a_6

和

a_1 a_2 a_3

a_4 a_5 a_6

是完全不同的。

该例中同样的数据元素，但是不同的关系构成了不同的（数据）结构。由此可见，数据结构不仅描述了在这个结构中有哪些数据元素，还刻画了这些元素之间的关系。

上述数据结构的定义仅是对操作对象的一种数学描述，换句话说，是从操作对象抽象出来的数学模型。结构定义中的关系是数据元素之间的逻辑关系，因此称之为数据的**逻辑结构**。讨论数据结构的目的是为了在计算机中实现对它的操作，因此还需研究如何在计算机中表示它。

数据结构在计算机中的表示称为**物理结构**，又称为**存储结构**。它是逻辑结构在存储器中的映像，包括数据元素的表示和关系的表示。数据结构的逻辑结构中的数据元素的映像方法是用二进制位（bit）的**位串**来表示数据元素，如一个十进制数126可以用一个位串1111110来表示，即$(126)_{10}=(176)_8=(1111110)_2$。一个字符“A”可以用其ASCII码01000001这样一个位串来表示，即$A=(65)_{10}=(101)_8=(01000001)_2$。

数据结构的逻辑结构中的关系的映像方法可以有顺序、链式、索引和散列等表示方法。

顺序映像（存储结构）的特点是借助元素在存储器中的相对位置表示数据元素之间的关系。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。

例1.6　有一数据结构$G = (D, R)$，其中$D = \{d_1, d_2, d_3, d_4, d_5\}$，$R = \{ \langle d_1, d_2\rangle, \langle d_2, d_3\rangle, \langle d_3, d_4\rangle, \langle d_4, d_5\rangle \}$。假定每个结点占一个存储单元，结点$d_1$放在200号单元中，则顺序方式存储的数据结构如图1-4所示。

200	d_1
201	d_2
202	d_3
203	d_4
204	d_5

图1-4　顺序方式存储的数据结构

链式映像（存储结构）是一种非顺序映像方法，借助指示元素存储地址的指针（Pointer）表示数据元素之间的逻辑关系，逻辑上相邻的元素其物理位置不要求相邻。链式存储结构通常借助于程序设计语言中的指针类型来实现。

例1.7　例1.6的数据结构的链式存储表示如图1-5所示。

图中的符号“NIL”表示空指针，该指针不是一个有意义的值，即不表示任何具体结点的单元地址。

从图1-4可以看出，在顺序方式实现的存储中所有的存储空间都被结点数据占用，它是一种紧凑结构。而在链式存储表示中一部分存储空间存放的是表示数据关系的附加信息，即指针，因此是一种非紧凑结构。

存储结构的**存储密度**定义为结构中数据本身所占的存储量和整个结构所占的存储量之比，即

$$d = \frac{\text{数据本身所占的存储量}}{\text{整个结构所占的存储量}}$$

可见，紧凑结构的存储密度为1，非紧凑结构的存储密度小于1。存储密度越大，则存储空间的利用率越高。但是非紧凑结构中存储的附加信息会给某些运算带来极大的方便，如在进行插入、删除等运算时链式存储结构比顺序存储结构就方便得多。其实非紧凑结构是牺牲了存储空间换取了机器时间。

	info	link
200	d_1	204
201		
202	d_3	203
203	d_4	207
204	d_2	202
205		
206		
207	d_5	NIL

图1-5 链式方式存储的数据结构

在索引存储方式中，线性结构中的数据结点被排成一个序列：d_1，d_2，…，d_n，每个结点d_i在序列里都有对应的位置数i，整个位置数就可以作为结点的索引。索引存储方式就是用结点的顺序号i来确定结点的存储地址。索引存储方式兼有静态和动态特性。

散列存储方式的主要思想是，在记录的存储地址和它的关键字之间建立一个确定的对应关系，使每个关键字和一个唯一的存储位置相对应。方法是根据设定的某个函数f(key)（散列函数）和处理冲突的方法将一组关键字映像到一个有限的连续地址集（区间）上，并以关键字在地址集中的“映像”作为记录在表中的存储位置，这种表称为散列表或哈希表。

一般数据结构的存储映像都采用这四种基本映像之一，或是它们的组合。同一个逻辑结构可以采用几种不同的映像方法，视问题的不同需求和具体应用而定。

在不同的程序设计环境中，存储结构有不同的描述方法。如果用高级语言进行程序设计，则可用高级程序设计语言提供的数据类型描述存储结构。如例1.4中的例子所示，以三个带有次序关系的整数表示一个长整数时，6524，3587，1496——a_1(6524)，a_2(3587)，a_3(1486)，可以用C语言中提供的整数数组类型定义长整数为：

```
typedef int Long_int[3];
```

1.3 抽象数据类型及其表示与实现

算法与数据结构作为一门计算机专业的专业基础课，本身也在不断的发展。一方面，发展各专门领域中特殊的数据结构，如多维图形数据结构；另一方面，从抽象数据类型的观点来讨论数据结构。

首先讨论**数据类型**（Data Type）的概念。数据类型是对数据的取值范围、数据元素之间的结构以及允许施加操作的一种总体描述。每一种计算机程序设计语言都定义有自己的数据类型。一般有整数、实数（浮点数）、字符、字符串、指针、数组、记录、类和文件等数据类型。例如，整数类型在计算机系统中通常用两个或四个字节表示。若采用两个字节，则整数表示范围在$-2^{15}\sim 2^{15}-1$，即$-32768\sim 32767$之间；若采用四个字节，则整数表示范围在$-2^{31}\sim 2^{31}-1$，即$-2147483648\sim 2147483647$之间。对整数类型的数据允许施加的操作（运算）通常有：单目取正取负运算，双目加、减、乘、除、取模等运算以及双目等于、不等于、大于、大于等于、小于、小于等于等关系（比较）运算以及赋值运算等。字符类型在计算机中通常用一个字节或两个字节表示，无符号表示范围分别在0～255或0～32767之间，能够分别表示至多256或32768种字符的编码。对字符类型的数据允许进行的操作主要为赋值和各种关系运算。字符串类型是

字符顺序排列的线性结构，每一个具体的字符串（其最大长度由具体的语言规定）都是字符串类型中的一个值，对字符串的操作有求串长度、串复制、串连接和串比较等。

按“值”的不同特性，数据类型可以分为**简单类型**和**结构类型**两大类。任一种简单类型中的每个数据都是无法再分割的整体，也称为**原子类型**。如一个整数、实数、字符、指针、枚举值、逻辑值等都是无法再分割的整体。任一种结构类型都是由简单类型数据按照一定的规则构造而成的，并且结构类型仍可以包含结构类型。所以一种结构类型中的数据（即结构数据）可以分解为若干个简单类型数据或结构类型数据，每个结构数据仍可再分。如数组就是一种结构类型，它由若干个分量组成，其中的每个分量可以是整数，也可以是数组等。数组中的每个数据（元素）都可以通过下标运算符直接访问。同样记录也是一种结构类型，它由固定个数的不同（也可以相同）类型的数据按线性结构排列而成，记录中的每个记录值包含有固定个数的不同类型数据，每个数据（域）都可以通过成员运算符直接访问。

无论是简单类型还是结构类型都有“型”和“值”的概念。一种数据类型中的任一数据称为该类型中的一个值（又称为实例），该值（实例）与所属数据类型具有完全相同的结构，数据类型所规定的操作就是在值上进行的。所以在一般的叙述中，并不明确指出是“型”还是“值”，应根据实际情况加以理解，例如，提到记录时，当讨论的是记录结构则认为是记录型，而当讨论的是具体的一条记录时则认为是记录值。

抽象数据类型（Abstract Data Type，ADT）是一个数学模型以及定义在该模型上的一组操作。抽象数据类型包含有一般数据类型的特征，但含义比一般数据类型更广、更抽象。一般数据类型通常由具体语言系统的内部定义，直接提供给用户定义数据并进行相应的运算，因此也称它们为系统预定义数据类型。抽象数据类型通常由用户根据已有的数据类型定义，包括定义其所含数据（数据结构）和在这些数据上所进行的操作。定义抽象数据类型就是定义其数据的逻辑结构和操作说明，而不必考虑数据的存储结构和操作的具体实现（即具体操作代码），从而使得抽象数据类型具有很好的通用性和可移植性，便于用任何一种语言，特别是面向对象的语言实现。

抽象数据类型和上面讨论的数据类型实质上是一个概念。例如，各个计算机系统都拥有的“整数”类型其实也是一个抽象数据类型，因为尽管它们在不同的处理器上实现的方法可能不同，但由于其定义的数学特性相同，所以在用户看来都是相同的。因此，“抽象”的意义在于数据类型的数学抽象特性。

使用抽象数据类型可以更容易地描述现实世界。例如，用线性表抽象数据类型描述学生成绩表，用树或图抽象数据类型描述遗传关系以及城市道路交通图等。抽象数据类型的特征是使用与实现相分离，实行封装和信息隐蔽。也就是说，在进行抽象数据类型设计时，把类型的定义与其实现分离开来。

按抽象数据类型的值的不同特性，抽象数据类型可分解为**原子类型**和**聚合类型**两大类。原子类型是其值不可分解的抽象数据类型，如整型数据类型。聚合类型又可分为**固定聚合类型**和**可变聚合类型**。其中，固定聚合类型的值由确定数目的成分按某种结构组成，如复数；而可变聚合类型的值的成分数目不确定，例如，可定义一个“有序整数序列”的抽象数据类型，其中序列的长度是可变的。

和数据结构的形式定义相对应，抽象数据类型可用如下三元组表示：

$$(D, R, P)$$

其中D是数据对象，即具有相同特性的数据元素的集合；R是D上的关系集合；P是对D的基本操作集合。

抽象数据类型的定义格式如下：

ADT 抽象数据类型名

{数据对象：<数据对象的定义>

数据关系：<数据关系的定义>

基本操作：<基本操作的定义>

}ADT 抽象数据类型名

其中的数据对象和数据关系可用伪代码描述，例如，线性表的抽象数据类型可定义如下：

ADT List

{数据对象：$D=\{a_i|a_i \in ElemSet,\ i=1,\ 2,\ \cdots,\ n,\ n \geqslant 0\}$

数据关系：$R=\{\langle a_{i-1},\ a_i\rangle|a_i,\ a_{i-1} \in D,\ i=2,\ \cdots,\ n\}$

基本操作：

线性表初始化：ListInit (L)；

求线性表的长度：ListLength (L)；

取表元素：ListGet(L，i)；

定位查找：ListLocate (L，x)；

清空线性表：ListClear (L)；

判空线性表：ListEmpty (L)；

求前驱：ListPrior (L，e)；

求后继：ListNext(L，e)；

插入：ListInsert(L，i，e)；

删除：ListDelete(L，i)；

}ADT List

抽象数据类型ADT中的基本操作的定义格式如下：

基本操作名（参数表）

初始条件：〈初始条件描述〉

操作结果：〈操作结果描述〉

“初始条件”描述了操作执行之前数据结构和参数应满足的条件。“操作结果”说明了操作正常完成之后，数据结构的变化状况和应返回的结果。若初始条件为空，则可省略。例如，上述线性表的抽象数据类型中的求线性表的长度操作可定义如下：

ListLength (L)

初始条件：线性表L存在。

操作结果：返回线性表L中所含元素的个数。

抽象数据类型可以通过固有数据类型来表示和实现，即利用处理器中已存在的数据类型来说明新的结构，用已经实现的一些操作组合来实现新的操作。本书采用介于伪代码和C语言之间的类C语言作为描述工具，有时也用伪代码描述一些只含抽象操作的抽象算法。

伪代码是一种用于描述算法的语言。它类似于计算机语言，但不是计算机语言；它是供人们阅读的，不是让计算机执行的。它可以使用源于某种计算机语言的赋值语句、条件语句和循环语句。在算法中我们也可以使用自然语言来描述某一步，只要这步显然能被执行完成。

伪代码语言介于高级程序设计语言和自然语言之间，它忽略高级程序设计语言中一些严格的语法规则与描述细节，因此它比程序设计语言更容易描述和被人理解，而比自然语言更接近程序设计语言。它虽然不能直接执行，但很容易被转换成计算机语言。类C语言是由伪代码和C语言组合而成的一个描述工具，采用了C语言的核心部分，并为描述方便进行了扩充。