21世纪 应用型本科计算机科学与技术专业规划教材

数据结构

周洪玉 邵晶波 主 编
许新民 马宪敏 刘宏 副主编
苏小红 主 审

清华大学出版社

21世纪应用型本科计算机科学与技术专业规划教材

数据结构

周洪玉　邵晶波　主　编
许新民　马宪敏　刘　宏　副主编
苏小红　主　审

清华大学出版社
北　京

内 容 简 介

本书根据计算机科学与技术及相关专业的培养目标来编写，突出实践动手能力的培养，使基础理论的教学以应用为目的。本书共分10章，第1章介绍数据结构基本概念；第2～4章主要介绍线性结构，分别是线性表、栈和队列、串；第5～7章介绍非线性结构，即多维数组和广义表、树、图；第8章和第9章介绍在数据处理中广泛使用的技术——查找和排序；第10章介绍文件。本书大部分算法都给出了完整的C语言源程序，各章均附有习题及参考答案。

本书可作为高等院校计算机科学与技术专业及相关专业的“数据结构”课程教材，也可作为计算机专业研究生入学考试、计算机认证考试用书，还可作为从事软件应用开发人员的参考用书。

图书在版编目(CIP)数据

数据结构/周洪玉，邵晶波主编. —北京：清华大学出版社，2011.8
(21世纪应用型本科计算机科学与技术专业规划教材)
ISBN 978-7-302-25945-9

Ⅰ. ①数… Ⅱ. ①周… ②邵… Ⅲ. ①数据结构－高等学校－教材 Ⅳ. ①TP311.12

中国版本图书馆CIP数据核字(2011)第122820号

责任编辑：索　梅　张为民
责任校对：李建庄
责任印制：何　芊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62795954，jsjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：23.5　**字　数**：583千字
版　　次：2011年8月第1版　　**印　　次**：2011年8月第1次印刷
印　　数：1～3000
定　　价：35.00元

产品编号：041413-01

前言

FOREWORD

数据结构是计算机科学与技术专业重要的专业基础课程与核心课程之一。多所高校一线教师的力量，根据多年的教学经验，在分析国内多种同类教材的基础上，博采众长，编写了这本书。

本书内容全面，案例丰富，几乎每个知识点都有对应的可运行的代码，所有实例代码都附有详细说明及运行效果截图，使读者在理解理论知识的基础上，加强实践认识，掌握解决实际问题的方法。本书共分 10 章，主要内容如下：

第 1 章介绍数据结构基本概念、数据的逻辑结构、存储结构和数据的运算和算法的描述工具、算法的评估。

第 2 章介绍线性表的概念、线性表的顺序存储结构和链式存储结构，以及在这两种存储结构上的基本运算。

第 3 章介绍栈和队列的概念、存储结构及运算。

第 4 章介绍串的概念、存储结构及运算。

第 5 章介绍数组和广义表的概念、存储结构及应用。

第 6 章介绍树和二叉树的概念、存储结构及遍历、线索化二叉树，还讲述了二叉树、树与森林三者之间的转换、树的应用。

第 7 章介绍图的概念、存储结构、遍历和图的生成树、图的应用等。

第 8 章介绍查找的概念、静态查找表、动态查找表及哈希表等。

第 9 章介绍排序的概念、各种排序的方法，包括插入排序、交换排序、选择排序、归并排序等。

第 10 章介绍文件的概述及文件的常见组织形式。

全书在内容选取上突出应用，在内容组织上循序渐进、由浅入深，在内容叙述上通俗易懂，按步骤讲解详尽，富于启发性。本书以 C 语言作为算法的描述语言，对于书中的实验和一些重要算法均给出了完整的 C 语言源程序，并且全部在 Visual C++环境中运行通过。另外，每章后面均附有习题及参考答案，方便加深和巩固所学的内容，提高学习效率。

本书由周洪玉、邵晶波主编，许新民、马宪敏、刘宏任副主编，苏小红主审。参加本书编写工作的还有杨喜林、郭俊凤、刘丽杰、张铁红、杨迎、郑立平、李申申等。具体分工是：杨喜林编写第 1 章，刘丽杰编写第 2 章和第 9 章，杨迎编写第 3 章，刘宏编写 4 章，许新民编写第

5章，邵晶波编写第6章，马宪敏编写第7章7.1节～7.4节和习题，李申申编写第7章7.5节和7.6节，郭俊凤编写第8章，张铁红编写第10章，郑立平编写附录A，全书由周洪玉和马宪敏统稿。

感谢哈尔滨工业大学博士生导师苏小红教授为本书进行全面审核，苏教授认真细致的工作保证了本书的质量。在编写过程中，我们力求做到严谨细致、精益求精，由于编写时间仓促，编者水平有限，书中疏漏和不妥之处在所难免，殷切希望读者和同行专家批评指正。

作　者

2011年6月

目 录

CONTENTS

第1章

绪　论

计算机的诞生是20世纪科学技术最伟大的成就之一。自1946年第一台电子计算机问世以来，计算机技术、计算机产业都在迅猛地发展，计算机的应用也以惊人的速度在普及。如今，计算机已深入到人类工作、学习、生活等各个领域。计算机的应用已不再局限于科学计算，而更多地应用于实时控制、信息管理、数据处理等非数值计算的处理工作；处理的对象也由简单的数值扩展到字符、表格、声音、图像等各种具有一定结构的数据。用计算机处理任何问题都离不开程序设计。为了编写出“好”的程序，必须分析待处理对象的特性及对象之间的关系。这就是“数据结构”这门学科形成和发展的背景。

1.1　数据结构的基本概念和术语

1.1.1　引言

通常情况下，在各种高级语言程序设计的学习过程中，解决一个实际问题的步骤是：分析实际问题；从问题中抽象出数学模型；设计解此数学模型的算法；编写程序；反复测试、调试程序直至得到正确结果。所谓抽象出数学模型，就是在分析实际问题时，从问题中提取操作的对象并找出对象之间存在何种关系，然后用数学语言进行描述。可以将数学模型理解成具体的代数方程等。然而，很多实际问题（非数值计算问题）无法用数学方程表示。下面几个典型实例的主要特点是对数据信息的存储与检索操作，并不是单纯的数值计算。

【例1.1】 单位员工档案的管理。

对单位员工的信息进行管理时，需要查询员工的档案簿。若利用计算机实现自动查询，则计算机处理的对象是档案中的员工信息。员工档案中一个员工的信息一般由员工号、姓名、性别、出生日期、学历、籍贯、参加工作时间等若干项组成，如表1.1所示。计算机主要对档案管理的操作包括查询、浏览、修改、插入、删除和汇总等。在这种文档管理的数学模型中，计算机处理的对象也就是一行的信息，行与行之间存在着一种简单的线性关系，这种关系称之为线性的数据结构。

表 1.1 某单位员工档案表

员工号	姓名	性别	出生日期	学历	籍贯	参加工作时间	…
001	王 刚	男	1980.5.12	本 科	山 东	2004.7.19	…
002	孙志刚	男	1965.8.17	专 科	黑龙江	1998.6.18	…
003	张莹莹	女	1977.12.8	研究生	沈 阳	1999.9.26	…
004	杨文晴	女	1979.4.20	本 科	吉 林	2002.3.15	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

【例 1.2】 大学学生的信息管理。

对大学学生的信息(见表 1.2)进行管理时,虽然可以采用例 1.1 中的二维表格形式将全校学生的名单列出来,但由于学生数量多,采用如图 1.1 所示的结构会更好。类似的问题还有人机对弈、体育竞技比赛等。通常,这种结构是非线性的,描述了学生所在的班级、专业和系。可以根据学生的信息,从根沿着某系某专业某班级快速地找到学生。如果将从学校查找某学生开始到找到某个学生结束的所有情况都画在一张图上,大家会看到一棵倒长着的“树”。“树根”是学校,“树叶”是所有可能要找的学生。假期学生到某些城市旅游,寻找一条最短路径,城市之间的道路构成一张无向图。对学生可以按平均成绩进行排序,可以按某些字段进行查找。这是某些非数值计算问题中的数学模型,称这种结构为树状结构。

表 1.2 学生信息表

学号	姓名	性别	出生日期	系	专业	家庭住址	平均成绩	…
990701	赵丽曼	女	1992.5.12	计算机	软件	北京	90	…
990702	张小飞	男	1990.8.17	计算机	网络	上海	85	…
990703	孙春艳	女	1991.12.8	计算机	软件	沈阳	94	…
990704	李 爽	女	1992.4.20	计算机	软件	西安	83	…
990705	石逸森	男	1991.10.6	计算机	软件	广州	65	…
990706	王 冬	女	1990.4.25	计算机	网络	重庆	63	…
990707	周明远	男	1988.11.9	计算机	网络	武汉	85	…
990708	钟少秋	男	1989.2.16	计算机	软件	哈尔滨	75	…
990709	张天波	男	1992.3.5	计算机	网络	长春	73	…
990710	刘一杰	男	1987.7.9	计算机	网络	南京	96	…

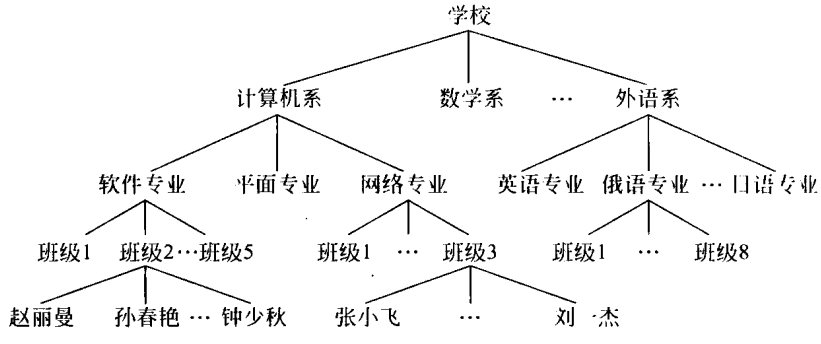

图 1.1 学生管理结构

【例 1.3】 GPS路径导航问题。

GPS导航系统是目前手机和导航仪上使用最广泛的系统，使用它可以求出某两个城市间的最短行驶路径。假设用图 1.2 中描述的关系来说明导航问题，图中的小圆圈表示一个城市，两个圆圈之间的连线表示两个城市之间的道路，连线上的数值表示要行驶的里程。当要从城市 1 去城市 2 时，要在两个城市之间有路的前提下，保证行驶的路程最短。这也是非数值计算问题中的数学模型，称这种结构为图状结构。

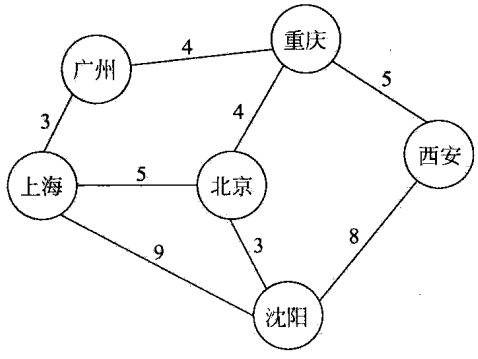

图 1.2 GPS路径导航问题

从上述 3 个例子可以看出，用计算机处理这些问题时，描述的非数值计算问题的数学模型不再是数学方程，计算机处理的对象之间存在着诸如表、树、图之类的数据结构。其他类型的数据结构都是由这 3 种基本类型派生出来的。

因此，在计算机科学中，数据结构是一门研究非数值计算的程序设计问题中，计算机的操作对象及其关系和运算等的学科，而且确保经过这些运算后所得到的新结构仍然是原来的结构类型。用计算机进行信息表示和处理，涉及两个问题：信息的表示和信息的处理。而信息的表示和组织又直接关系到处理信息的程序的效率。随着计算机的普及，信息量不断增加，信息范围日益拓宽，使许多系统程序和应用程序的规模很大，结构又相当复杂。因此，为了编写出一个“好”的程序，必须分析待处理对象的特征及各对象之间存在的关系，这就是数据结构这门课所要研究的问题。众所周知，计算机的程序是对信息进行加工处理。在大多数情况下，这些信息并不是没有组织的，信息（数据）之间往往具有重要的结构关系，这就是数据结构的内容。数据的结构，直接影响算法的选择和效率。

“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 年美国唐·欧·克努特教授开创了数据结构的最初体系，他所著的《计算机程序设计技巧》第一卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。70 年代初，“数据结构”作为一门独立的课程开始进入大学课堂。

数据结构随着程序设计的发展而发展。程序设计经历了 3 个阶段：无结构阶段、结构化阶段和面向对象阶段，相应地，数据结构的发展也经历了 3 个阶段。

1. 无结构阶段

20 世纪 40—60 年代，计算机的应用主要针对科学计算，程序设计技术以机器语言/汇编语言为主，程序处理的数据是纯粹的数值，数据之间的关系主要是数学公式或数学模型。这一阶段，在人类的自然语言与计算机编程语言之间存在着巨大的鸿沟，程序设计属于面向计算机的程序设计，设计人员关注的重心是使程序尽可能地被计算机接受并按指令正确执行，至于程序能否让人理解并不重要。

2. 结构化阶段

20 世纪 60—80 年代，计算机开始广泛应用于非数值处理领域，数据表示成为程序设计

的重要问题，人们认识到程序设计规范化的重要性，提出了程序结构模块化，并开始注意数据表示与操作的结构化。数据结构及抽象数据类型就是在这种情况下形成的。数据结构概念的引入，对程序设计的规范化起到了重大作用。图灵奖获得者沃思给出了一个著名的公式：数据结构＋算法＝程序。从这个公式可以看到，数据结构和算法是构成程序的两个重要的组成部分，一个软件系统通常是以一个或几个关键数据结构为核心而组织的。

随着软件系统的规模越来越大、复杂性不断增加，人们不得不对结构化技术重新评价。由于软件系统的实现依赖于关键数据结构，如果这些关键数据结构的一个或几个有所改变，则涉及整个系统，甚至导致整个系统彻底崩溃。

3. 面向对象阶段

面向对象技术始于 20 世纪 80 年代初，是目前最流行的程序设计技术。在面向对象技术中，问题的相关实体被视为一个对象，对象由属性和方法构成，属性用以描述实体的状态或特征，方法用以改变实体的状态或描述实体的行为。一组具有相同属性和方法的对象的集合抽象为类，每个具体的对象都是类的一个实例。例如，“教师”是一个类，“张老师”、“杨老师”等对象都是“教师”类的实例。

由于对象(类)将密切相关的属性(数据)和方法(操作)定义为一个整体，从而实现了封装和信息隐藏。使用类时，无需了解其内部的实现细节，一旦数据(结构)修改了，只需修改类内部的局部代码，软件系统的其余部分无需修改。

数据结构主要强调两个方面的内容：数据之间的关系；针对这些关系的基本操作。这两个方面实际上蕴含着面向对象的思想：类重点描述实体的状态与行为，而数据结构重点描述数据之间的关系及其基本操作，数据及其相互关系构成了对实体状态的描述，针对数据元素之间关系的操作构成了对实体行为的描述。由此可见，类与数据结构之间具有对应关系，如图 1.3 所示。

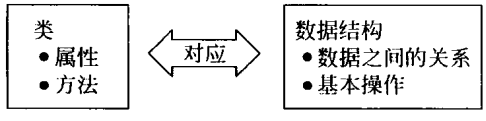

图 1.3　类与数据结构之间的对应关系

值得注意的是，数据结构的发展并未终结。一方面，数据结构将继续随着程序设计的发展而发展；另一方面，面向各专门领域的数据结构得到研究和发展，各种实用的高级数据结构被研究出来，各种空间数据结构也在探索中。

“数据结构”在计算机科学中是一门综合性的专业基础课。它是介于数学、计算机硬件和计算机软件三者之间的一门核心课程，这一门课的内容不仅是一般程序设计(特别是非数值性程序设计)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的重要基础。

1.1.2　数据结构有关概念及术语

数据(Data)是客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并能被计算机程序识别和处理的符号集合。数据可以分为两大类：一类是整数、实数等数值数

据；另一类是图形、图像、声音、文字等非数值数据。

数据是计算机程序加工的“原料”，例如，编译程序加工的数据是源程序；单位员工档案的管理程序加工的数据是档案登记表；大学学生的信息管理程序加工的数据是学生信息表。

数据元素(Data Element)是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。构成数据元素的不可分割的最小单位称为**数据项**(Data Item)。例如，例 1.1 中的员工档案，每个员工的信息(一行)就是一个数据元素，而档案中的员工号、姓名、出生日期等是数据项，一个数据元素由若干个数据项组成。数据元素是讨论数据结构时涉及的最小数据单位，其中的数据项一般不予考虑。

数据元素具有广泛的含义，一般来说，能独立、完整地描述问题世界的一切实体都是数据元素。例如，人机对弈中的棋盘格局、学生管理中的某个系、一年中的 4 个季节，甚至一次授课过程、一场乒乓球比赛都是数据元素。数据元素又称为元素、结点、顶点或记录。

数据对象(Data Object)是具有相同性质的数据元素的集合，是数据的子集，如字母数据对象是集合 $\Sigma=\{'A','B',\cdots,'Z'\}$。在解决具体问题时处理的数据元素通常具有相同的性质，例如，学生的信息管理中每个数据元素具有相同数目和类型的数据项，所有数据元素(学生的信息)的集合就构成了一个数据对象。在不产生混淆的情况下，将数据对象简称为数据。

数据结构(Data Structure)是指相互之间存在一种或多种特定关系的数据元素的集合。

按照用户视点的不同，数据结构分为逻辑结构和存储结构。

数据的**逻辑结构**(Logical Structure)是指数据元素之间逻辑关系的整体。所谓逻辑关系是指数据元素之间的关联方式或邻接关系。从 1.1.1 节中的 3 个例子可以看出，在任何问题中，数据元素之间并不是孤立的，而是存在着某种关系，这种数据元素间的关系称为**结构**(Structure)。根据数据元素之间逻辑关系的不同，数据结构可分为以下 4 类：

(1) 集合：数据元素之间除“属于同一个集合”外，没有任何关系。

(2) 线性结构：数据元素之间存在着一对一的线性关系。

(3) 树状结构：数据元素之间存在着一对多的层次关系。

(4) 图状结构(网状结构)：数据元素之间存在着多对多的任意关系。

树状结构和图状结构也称为非线性结构。

数据的逻辑结构常用逻辑结构图来描述，其描述方法是：将每一个数据元素看做一个结点，用圆圈表示，元素之间的逻辑关系用结点之间的连线表示，如果强调关系的方向性，则用带箭头的连线表示关系。图 1.4 描述了 4 种基本数据结构的逻辑结构。

数据的逻辑结构属于用户视图，是针对问题而言的，反映了数据内部以何种方式构成。为了区别于数据的存储结构，常常将数据的逻辑结构称为数据结构。

数据的**存储结构**(Storage Structure)又称为**物理结构**，是数据及其逻辑结构在计算机中的表示，也就是说，存储结构除了存储数据元素之外，

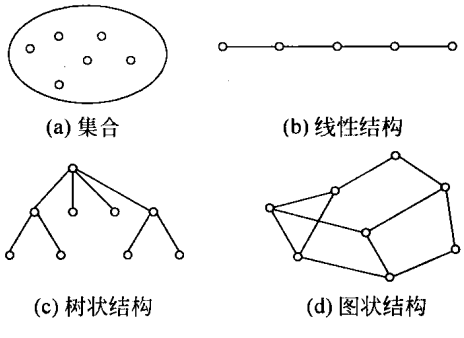

图 1.4 4 种基本数据结构的逻辑结构

还必须将存储数据元素之间的逻辑关系隐式或显式地表示出来。存储结构通常有两种：顺序存储结构和链式存储结构。

顺序存储结构的基本思想是：用一组连续的存储单元依次存储数据元素，数据元素之间的逻辑关系是由元素的存储位置来表示的，如线性结构(A,B,C) 的顺序存储示意图如图 1.5 所示。

链接存储结构的基本思想是：用一组任意的存储单元存储数据元素，数据元素之间的逻辑关系是用指针来表示的，如线性表(A,B,C)的链式存储示意图如图 1.6 所示。

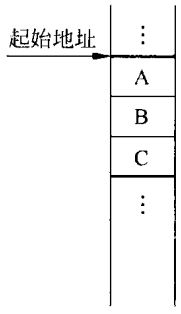

图 1.5　线性表的顺序存储示意图

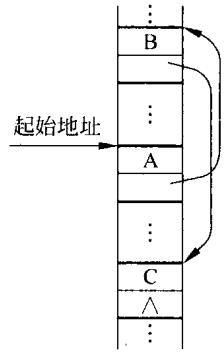

图 1.6　线性表的链式存储示意图

数据的存储结构是数据在计算机中具体实现的视图，是面向计算机的，其基本目标是将数据及其逻辑关系存储到计算机的内存中，用户无法看到。在高级语言中都有的“一维数组”类型描述的是数据的顺序存储结构，C 语言所提供的“指针”描述的是链式存储结构。

数据的逻辑结构和存储结构是密切相关的两个方面。一般来说，一种数据的逻辑结构可以用多种存储结构来存储，而采用不同的存储结构，其数据处理的效率往往是不同的。

1.1.3　数据结构和抽象数据类型

数据类型(Data Type)是一组性质相同的值的集合以及定义于这个值集上的一组操作的总称。数据类型规定了该类型数据的取值范围和对这些数据所能采取的操作。例如，C 语言中的整型变量可以取的值是机器所能表示的最小负整数和最大正整数之间的任何一个整数，这是整型数据类型所取得值的集合；允许的操作有＋、－、＊、/、%、<、<＝、>、>＝、＝＝、!＝等，这是整型数据类型在值的集合上所进行的操作的集合。

在高级程序设计语言中，根据“值”的不同特性，可以将数据类型分为原子类型和结构类型两类：原子类型是非结构的，值是不可分解的，如 C 语言中的基本类型(整型、实型、字符型、枚举型)、指针类型、空类型等；结构类型的值是由若干成分按一定结构组成，是可以分解的，如数组由若干分量组成，每个分量可以是整型等基本数据类型，也可以是数组等。

抽象(Abstract)是从众多的事物中抽取出共同的、本质性的特征，而舍弃其非本质的特征。例如，芒果、苹果、香蕉、菠萝、桃子等共同的特性就是水果。得出水果概念的过程，就是一个抽象的过程。要抽象，就必须进行比较，没有比较就无法找到共同的部分。抽象化主要是为了使复杂度降低，以得到论域中较简单的概念，好让人们能够控制其过程或以综观的角度来了解许多特定的事态。