新世纪高等院校精品教材

数据结构与数据库技术

孙志锋 徐镜春 厉小润 编著

浙江大學出版社

数据结构与数据库技术

孙志锋　徐镜春　厉小润　编著

浙江大學出版社

内 容 简 介

本书主要内容包括两大部分:第一部分为数据结构,包括线性表、栈和队、串、数组、树、图等,以及排序和查找等操作;第二部分为数据库技术,包括数据库概论、数据库技术基础、关系数据库基本理论、数据库设计、关系数据库标准语言 SQL 等。

本书适合非计算机专业的本、专科教材,也可供自学计算机基础知识的读者参考。

图书在版编目(CIP)数据

数据结构与数据库技术 / 孙志锋等编著. —杭州:浙江大学出版社,2004.8
新世纪高等院校精品教材
ISBN 7-308-03800-9

Ⅰ.数... Ⅱ.孙... Ⅲ.①数据结构-高等学校-教材②数据库系统-高等学校-教材 Ⅳ.TP311.1

中国版本图书馆 CIP 数据核字(2004)第 073333 号

出版发行 浙江大学出版社
(杭州浙大路 38 号 邮政编码 310027)
(E-mail:zupress@mail.hz.zj.cn)
(网址:http://www.zjupress.com)
责任编辑 杜希武 李文启
排 版 浙江大学出版社电脑排版中心
印 刷 德清第二印刷厂
开 本 787mm×1092mm 1/16
印 张 20
字 数 512 千字
版 印 次 2004 年 8 月第 1 版 2004 年 8 月第 1 次印刷
书 号 ISBN 7-308-03800-9/TP·260
定 价 30.00 元

序言

数据结构和数据库技术是计算机程序设计的重要理论技术基础，它不仅是高等学校计算机科学与技术类专业学生必修的两门专业基础课程，而且已成为其他理工专业的热门选修课。目前，有关数据结构和数据库技术的书籍很多。随着课程建设的改革、课时的缩减，如何能使学生在有限的课时里更好地掌握这两门课程，并能在实际的软件开发过程中自觉地应用，一直是摆在广大教师面前的课题。

本书的作者长期从事计算机软件技术基础的教学工作，他们根据教学大纲的要求，结合目前的教学实际情况，认真编写了《数据结构与数据库技术》这本教材。我通读了全书，发现本书有许多独到之处，概括起来有以下几个方面：

1. 本书在介绍数据结构的基本运算时，不是仅仅局限于算法思想，而是着眼于程序的实现过程。书中给出的用C++语言编制的各种数据结构算法源程序都真正上机调试通过，便于读者更深刻地领会数据结构的内涵，弥补了其他有关数据结构书籍的不足。

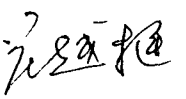
2. 作者在介绍数据库技术的基本原理和方法的时候，以一个实际的研究课题为主线，深入浅出地讨论了数据库的应用技术，这样有助于学生从实践的角度来深入理解数据库的原理，这种写法在同类教科书中是比较少见的。

3. 本书首先详细讲述数据结构的相关知识，而在介绍数据库设计的时候，又把软件工程的方法贯穿于其中，对于非计算机专业的计算机软件基础课程而言，这样的内容安排是非常适合的。

4. 全书对概念、原理的阐述不仅非常准确、精炼，而且通俗易懂。

我认为，本书对从事计算机软件教学和开发的人员来说是一部不可多得的好书。能够为本书作序我感到非常荣幸。希望本书能够成为广大读者的良师益友。

浙江大学计算机学院　教授、博士生导师
浙江大学计算机学院　副院长
浙江大学软件学院　常务副院长



2004年8月8日于浙大求是园

前言

随着个人计算机和 Internet 的迅猛发展，以计算机科学技术为核心的信息技术正在深刻地改变着人们的工作方式、生活方式和思维方式。有人预计 21 世纪的计算机软件业将成为全球高科技产业发展的最主要的推动力。因此，作为软件设计技术的理论基础，“数据结构与数据库技术”不仅仅是计算机科学技术学科的核心课程，也是所有应用计算机的其它理工学科需要掌握的课程。

通过对本课程的学习，使学生掌握各种数据结构的基本原理和算法，提高对复杂程序的设计技能，培养良好的程序设计习惯；使学生掌握数据库系统的理论基础、结构化查询语言 SQL 及数据库系统的开发技术。

本书是为浙江大学电气工程学院“数据结构与数据库技术”课程编写的教材。在内容表达上，既注重原理又重视实践，并配有大量解释详尽的例题。数据结构部分每章的算法都在 Visual C++ 6.0 上调试通过，有助于读者对课本内容的正确理解和上机实验。本书语言流畅，通俗易懂，内容循序渐进，可以作为高等院校非计算机专业本科生、专科生及成人教育或高等职业专科院校的教材，也可供广大从事计算机软件与应用工作的科技人员及自学考试者参考。

本书的数据结构部分由孙志锋编写，数据库技术部分由厉小润编写，徐镜春对全书的算法进行了上机调试，并对全书的内容进行了仔细的校核，最后由孙志锋对全书进行了统稿。

本书的出版得到了浙江大学各有关部门的领导的大力支持，并受到许多同行专家的指点，在此一并表示衷心感谢。

由于作者水平有限，书中难免有一些不妥之处，恳请读者批评指正。

作　者

2004 年 6 月于浙大求是园

目录

第一部分　数据结构

第二部分　数据库技术

第一部分

数据结构

第1章

绪　论

计算机科学是一门研究数据表示和数据处理的科学。数据是信息在计算机中的表示形式，它是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算或数据处理、过程控制，还是对文件的存储和检索等，都是对数据进行加工处理的过程。因此，要设计出一个结构好效率高的数据加工处理程序，必须研究数据的组织特性、数据间的相互关系及其对应的存储表示，并利用这些特性和关系设计出相应的算法和程序。数据结构这门学科就是研究这些问题的。

1.1　数据结构的概念

数据结构是计算机科学与技术专业的专业基础课，是十分重要的核心课程。所有的计算机系统软件和应用软件都要用到各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，仅掌握几种计算机程序设计语言是难以应付众多复杂的课题的。要想有效地使用计算机，充分发挥计算机的性能，还必须学习和掌握好数据结构的有关知识。打好“数据结构”这门课程的扎实基础，对于学习计算机专业的其他课程，如操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是十分有益的。

1.1.1　为什么要学习数据结构

在计算机发展的初期，人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行调试、测试，直至得到最终的解答。例如，求解梁架结构中应力的数学模型的线性方程组，该方程组可以使用迭代算法来求解。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据，所以程序设计者的主要精力是集中于程序设计的技巧上，而无须重视数据结构。随着计算机应用领域的扩大和软、硬件的发展，非数值计算问题越来越显得重要。据统计，当今处理非数值计算性问题占用了90%以上的机器时间。这类问题涉及到的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要

设计出合适的数据结构，才能有效地解决问题。下面所列举的就是属于这一类的具体问题。

【例 1】 学生信息检索系统。当我们需要查找某个学生有关情况的时候，或者想查询某个专业或年级的学生有关情况的时候，只要我们建立了相关的数据结构，按照某种算法编写了相关程序，就可以实现计算机自动检索。由此，可以在学生信息检索系统中建立一张按学号顺序排列的学生信息表和分别按姓名、专业、年级顺序排列的索引表，如图 1.1 所示。这四张表便是学生信息检索的数学模型，计算机的主要操作便是按照某个特定要求（如给定姓名）对学生信息表进行查询。

诸如此类的还有电话自动查号系统、考试查分系统、仓库库存管理系统等。在这类文档管理的数学模型中，计算机处理的对象之间通常存在着的是一种简单的线性次序关系，这类数学模型可称为线性数据结构。

学号	姓名	性别	专　业	年　级
980001	吴承志	男	计算机科学与技术	1998 级
980002	李淑芳	女	信息与计算科学	1998 级
990301	刘　丽	女	数学与应用数学	1999 级
990302	张会友	男	信息与计算科学	1999 级
990303	石宝国	男	计算机科学与技术	1999 级
000801	何文颖	女	计算机科学与技术	2000 级
000802	赵胜利	男	数学与应用数学	2000 级
000803	崔文婧	男	信息与计算科学	2000 级
010601	刘　丽	女	计算机科学与技术	2001 级
010602	魏永鸣	男	数学与应用数学	2001 级

(a)学生信息表

崔文婧	8
何文颖	6
李淑芳	2
刘　丽	3,9
石宝国	5
魏永鸣	10
吴承志	1
赵胜利	7
张会有	4

(b)姓名索引表

计算机科学与技术	1,5,6,9
信息与计算科学	2,4,8
数学与应用数学	3,7,10

(c)专业索引表

2000 级	6,7,8
2001 级	9,10
1998 级	1,2,3
1999 级	4,5

(d)年级索引表

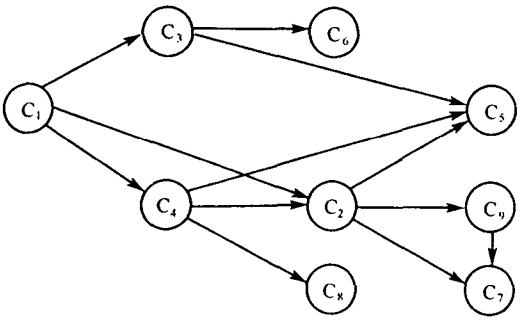
图 1.1　学生信息查询系统中的数据结构

【例 2】 教学计划编排问题。一个教学计划包含许多课程，在教学计划包含的许多课程之间，有些必须按规定的先后次序进行排列，有些则没有次序要求。即有些课程之间有先修和后续的关系，有些课程可以任意安排次序。这种各个课程之间的次序关系可用一个称作图的

数据结构来表示，如图 1.2 所示。有向图中的每个顶点表示一门课程，如果从顶点 v_i 到 v_j 之间存在有向边$<v_i, v_j>$，则表示课程 i 必须先于课程 j 进行。

课程编号	课程名称	先修课程
C_1	计算机导论	无
C_2	数据结构	C_1, C_4
C_3	汇编语言	C_1
C_4	C 程序设计语言	C_1
C_5	计算机图形学	C_2, C_3, C_4
C_6	接口技术	C_3
C_7	数据库原理	C_2, C_9
C_8	编译原理	C_4
C_9	操作系统	C_2

(a)计算机专业的课程设置



(b)表示课程之间优先关系的有向图

图 1.2　教学计划编排问题的数据结构

由以上两个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如表、图之类的数据结构。因此，可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构的目的是为了了解计算机处理对象的特性，将实际问题中所涉及的处理对象在计算机中表示出来并对它们进行处理。与此同时，通过算法训练来提高学生的思维能力，通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

1.1.2　有关概念和术语

在系统地学习数据结构知识之前，首先应对一些基本概念和术语赋予确切的含义。

数据(Data)：是能够被计算机识别、存储和加工处理的信息的载体。它是计算机程序加工的原料。计算机科学中，所谓数据就是计算机加工处理的对象，它可以是数值数据，也可以是非数值数据。数值数据是一些整数、实数或复数，主要用于工程计算、科学计算和商务处理等；非数值数据包括字符、文字、图形、图像、语音等。

数据元素(Data Element)：是数据的基本单位。在不同的情况下，数据元素又可称为元素、结点、顶点、记录等。例如，学生信息检索系统中学生信息表中的一个记录，教学计划编排

问题中的一个顶点等,都被称为一个数据元素。

数据对象(Data Object)或**数据元素类**(Data Element Class)L 是具有相同性质的数据元素的集合。在某个具体问题中,数据元素都具有相同的性质(元素值不一定相等),属于同一数据对象(数据元素类),数据元素是数据元素类的一个实例。例如,在交通咨询系统的交通网中,所有的顶点是一个数据元素类,顶点 A 和顶点 B 各自代表一个城市,是该数据元素类中的两个实例,其数据元素的值分别为 A 和 B。

数据结构(Data Structure):指的是数据之间的相互关系,即数据的组织形式。虽然至今没有一个关于数据结构的标准定义,但它一般包括以下三方面的内容:

(1)数据元素之间的逻辑关系,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存储器内的表示,也称为数据的存储结构(Storage Structure)。

(3)数据的运算,即对数据施加的操作。

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,是独立于计算机的。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。数据的存储结构是逻辑结构在计算机存储器里的实现(亦称为映象),它是依赖于计算机的,对机器语言而言,存储结构是具体的,但我们只在高级语言的层次上来讨论存储结构。数据的运算是定义在数据的逻辑结构上的,每种逻辑结构都有一个运算的集合。例如,最常用的运算有:检索,插入,删除,更新,排序等。这些运算实际上是在抽象的数据上所施加的一系列抽象的操作,所谓抽象的操作,是指我们只知道这些操作是"做什么",而无须考虑"如何做"。只有确定了存储结构之后,我们才考虑如何具体实现这些运算。本书中讨论的数据运算,均在高级语言的层次上用相应的算法来实现。

在不会产生混淆的前提下,我们常将数据的逻辑结构简称为数据结构。

数据的逻辑结构有以下两大类:

1) 线性结构

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,并且所有结点都最多只有一个直接前趋和直接后继。

2) 非线性结构

非线性结构的逻辑特征:一个结点可能有多个直接前趋和直接后继。

数据的存储结构可用以下四种基本的存储方法得到:

1) 顺序存储方法:把逻辑上相邻的结点存储在物理位置上相邻的存储单元里,结点间的逻辑关系由存储单元的邻接关系来体现。

2) 链接存储方法:该方法不要求逻辑上相邻的结点在物理位置上也相邻,结点间的逻辑关系由附加的指针字段表示。

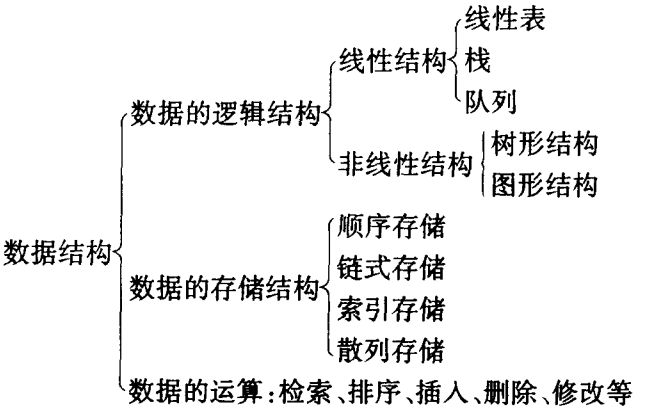
3) 索引存储方法:在存储结点信息的同时,还建立附加的索引表。

4) 散列存储方法:根据结点的关键字直接计算出该结点的存储地址。

同一逻辑结构可有不同的存储结构;同理在给定了数据的逻辑结构和存储结构之后,按定义的运算集合及其运算的性质不同,也可能导致完全不同的数据结构。例如,线性表是一种逻辑结构,若采用顺序方法的存储表示,则为顺序表;若采用链接方法的存储表示,则为链表;若采用散列方法的存储表示,则为散列表;若对线性表上的插入、删除运算限制在表的一端进行,则为栈;若对插入限制在表的一端进行,而删除限制在表的另一端进行,则为队列。

综上所述,我们可以将**数据结构**定义为:按某种数据关系组织起来的一批数据,应用计算机语言,可按一定的存储表示方式把它们存储在计算机的存储器中,并在该数据中定义了一个运算的集合。

上述这些基本概念的相互关系归纳如下:



1.2 算法和算法分析

算法与数据结构的关系紧密,在算法设计时,先要确定相应的数据结构,而在讨论某一种数据结构时,也必然会涉及实现一定功能的算法。下面就从算法特性、算法描述、算法性能分析与度量等三个方面对算法进行介绍。

1.2.1 算法特性

算法(Algorithm)是对特定问题求解步骤的一种描述,是指令的有限序列。其中每一条指令表示一个或多个操作。一个算法应该具有下列特性:

(1)有穷性:一个算法必须在有穷步之后结束,即必须在有限时间内完成。

(2)确定性:算法的每一步必须有确切的定义,无二义性。

(3)可行性:算法中的每一步都可以通过已经实现的基本运算的有限次执行得以实现。

(4)输入性:一个算法具有零个或多个输入,这些输入取自特定的数据对象集合。

(5)输出性:一个算法具有一个或多个输出,这些输出同输入之间存在某种特定的关系。

算法的含义与程序十分相似,但又有区别。一个程序不一定满足有穷性。例如操作系统,只要整个系统不遭破坏,它将永远不会停止,即使没有作业需要处理,它仍处于动态等待中。因此,操作系统不是一个算法。另一方面,程序中的指令必须是机器可执行的,而算法中的指令则无此限制。算法代表了对问题的求解方法和步骤,而程序则是算法在计算机上的特定的实现。一个算法若用程序设计语言来描述,则它就是一个程序。

算法与数据结构是相辅相成的。解决某一特定类型问题的算法可以选定不同的数据结构,而且选择恰当与否直接影响算法的效率;反之,一种数据结构的优劣由各种算法的执行效率来体现。

要设计一个好的算法通常要考虑以下的要求:

(1)正确:算法的执行结果应当满足预先规定的功能和性能要求。

(2)可读:一个算法应当思路清晰、层次分明、简单明了、易读易懂。

(3)健壮:当输入不合法数据时,应能作适当处理,不至引起严重后果。

(4)高效:有效使用存储空间和有较高的时间效率。

1.2.2 算法描述

算法可以使用各种不同的方法来描述。

最简单的方法是使用自然语言。用自然语言来描述算法的优点是简单且便于人们对算法的阅读。缺点是不够严谨。

可以使用程序流程图、N-S 图等算法描述工具。其特点是描述过程简洁、明了。

用以上两种方法描述的算法不能够直接在计算机上执行,若要将它转换成可执行的程序还有一个编程的问题。

可以直接使用某种程序设计语言来描述算法,不过直接使用程序设计语言并不容易,而且不太直观,常常需要借助于注释才能使人看明白。

为了解决理解与执行这两者之间的矛盾,人们常常使用一种称为伪码语言的描述方法来进行算法描述。伪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高级语言。

由于绝大部分读者都学过 C 语言,故本书的算法均用 C++来描述。

1.2.3 算法性能分析与度量

求解同一个问题,可以有许多不同的算法,那么如何来评价这些算法的好坏呢?

显然,选用的算法首先应该是"正确的"。此外,主要考虑如下三点:

(1)执行算法所耗费的时间;

(2)执行算法所耗费的存储空间,其中主要考虑辅助存储空间;

(3)算法应易于理解,易于编码,易于调试等等。

当然我们希望选用一个所占存储空间小、运行时间短、其他性能也好的算法。然而,实际上很难做到十全十美。原因是上述要求有时相互抵触。要节约算法的执行时间往往要以牺牲更多的空间为代价;而为了节省空间又可能要以更多的时间作代价。因此我们只能根据具体情况有所侧重。若该程序使用次数较少,则力求算法简明易懂,易于转换为上机的程序。对于反复多次使用的程序,应尽可能选用快速的算法。若待解决的问题数据量极大,机器的存储空间较小,则相应算法主要考虑如何节省空间。本书主要讨论算法的时间特性,偶尔也讨论空间特性。

我们可以从一个算法的时间复杂度与空间复杂度来评价算法的优劣。

1. 时间复杂度

一个算法所耗费的时间,应该是该算法中每条语句的执行时间之和,而每条语句的执行时间是该语句的执行次数(也称为频度(Frequency Count))与该语句一次执行所需时间的乘积。当我们将一个算法转换成程序并在计算机上执行时,其运行所需要的时间取决于下列因素: