国家精品课程主讲教材

数据结构：思想与实现

翁惠玉　俞勇　编著

高等教育出版社
Higher Education Press

国家精品课程主讲教材

数据结构：思想与实现

翁惠玉　俞　勇　编著

高等教育出版社

内容提要

本书为国家精品课程“数据结构”的主讲教材。本书条理清晰，严格按照线性结构、树形结构、集合结构和图形结构的次序来组织编写。除了常规的数据结构内容之外，还介绍了一些高级的数据结构，如红黑树、AA 树和跳表等，并提供了大量的数据结构应用实例。让读者在学习数据结构的同时，逐步了解为什么要学习数据结构，了解数据结构对计算机专业的重要性。

本书内容翔实，既注重数据结构和算法的原理，又十分强调和程序设计课程的衔接。在讲授数据结构的同时，不断加强学生对程序设计的理解。书中的算法都有完整的 C++ 实现。这些程序结构清晰，构思精巧。所有的程序都在 VC 6.0 环境下编译通过，并能正确运行。它们既是学习数据结构和算法的示例，也是学习 C++ 程序设计很好的示例。

本书可作为高等学校计算机及相关专业数据结构课程教材，也可作为参加计算机专业硕士研究生入学考试的参考用书。

图书在版编目(CIP)数据

数据结构：思想与实现/翁惠玉，俞勇编著. —北京：
高等教育出版社，2009.8
ISBN 978-7-04-027783-8

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

中国版本图书馆 CIP 数据核字(2009)第 104080 号

策划编辑	刘 艳	**责任编辑**	刘 艳	**封面设计**	于文燕
版式设计	史新薇	**责任校对**	殷 然	**责任印制**	尤 静

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	**咨询电话**	400-810-0598
邮政编码	100120	**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	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	**畅想教育**	http://www.widedu.com
开 本	850×1168 1/16	**版 次**	2009 年 8 月第 1 版
印 张	27	**印 次**	2009 年 8 月第 1 次印刷
字 数	570 000	**定 价**	34.40 元

物料号 27783-00

前　言

数据结构是计算机专业最基础，也是最重要的课程之一。它和程序设计一起为计算学科其他后继课程的学习奠定了基础。

数据结构和程序设计是关系非常密切的课程。在教学安排中，通常在程序设计之后接着就是数据结构。在学习数据结构时，学生对程序设计还不是非常熟练。通常学生的难点不在于对数据结构的理解，而在于如何用特定的程序设计语言来实现这些数据结构，特别是如何按照面向对象的思想将一个个数据结构设计成一个个类。本书十分强调和程序设计课程的衔接。在讲授数据结构的同时，不断加强学生对程序设计的理解。书中的算法都有完整的 C++ 实现。这些程序结构清晰，构思精巧。所有的程序都在 VC 6.0 环境下编译通过，并能正确运行。它们既是学习数据结构和算法的示例，也是学习 C++ 程序设计很好的示例。

本书既适合用过程化的方法讲授，也适合用面向对象的方法讲授。除第 1 章之外，其余各章的结构基本上是一致的。每 1 章介绍一个数据结构，分成几个部分描述。第一部分介绍数据结构所处理的逻辑结构以及该数据结构的常用操作。第二部分介绍数据结构的各种实现方法，本书采用了自然语言加伪代码的方式描述这些实现方法。第三部分介绍如何将该数据结构封装成类，用 C++ 语言描述。第 4 部分介绍 C++ 中对应于该数据结构的工具，告诉读者如何应用现有的工具。最后一部分介绍该数据结构的应用，让读者进一步了解数据结构的重要性。

在授课过程中，教师可以按照教学实际需要选择内容。如果采用过程化的方法讲课，则可以选用每一章的第一、第二部分和最后一部分。如果采用面向对象的方式，可以增加第三部分。如果读者经常用 C++ 编程，可以介绍第四部分。

本书条理清晰，严格按照线性结构、树形结构、集合结构和图形结构的次序来组织编写。除了第 1 章以外，其余各章被分成五大部分：线性结构、树形结构、集合结构、图形结构和算法设计。本书共分 15 章。

第 1 章是基础知识，介绍什么是数据结构，什么是算法；回顾面向对象的程序设计方法；最后还介绍了本书的使用方法。

第 2、3、4 章属于第一部分，主要讨论线性结构。第 2 章介绍线性表，第 3 章介绍栈，第 4 章介绍队列。

第 5、6 章属于第二部分，主要讨论树形结构。第 5 章介绍了树和二叉树，以及二叉树的一些应用。第 6 章介绍了基于树实现的优先级队列。

第三部分讨论集合结构，由第 7、8、9、10、11 章组成。这一部分是根据集合的两个基本

操作,即查找和排序组织的。第7章介绍了集合的基本概念及静态查找表。第8章介绍了支持插入和删除操作的动态查找表。第9章介绍了专用于集合的一个数据结构:散列表。第10章介绍了排序操作。第11章介绍了基于集合操作的、用于处理等价类的工具:不相交集。

第四部分讨论图形结构,由第12、13、14章组成。第12章介绍了图的基本概念、基本操作及实现方法。第13、14章分别介绍图中的两个重要操作:最小生成树和最短路径问题。

最后一部分由第15章组成,介绍了基本的算法设计方法。主要介绍了枚举法、贪婪法、分治法、动态规划和随机算法,使读者在遇到一些没有现成算法的问题时知道如何着手解决问题。

本书内容丰富,除了常规的数据结构内容之外,还介绍了一些高级的数据结构,如红黑树、AA树、不相交集和跳表等,并提供了大量的数据结构应用实例,让读者在学习数据结构的同时,逐步了解为什么要学习数据结构,了解数据结构对计算机专业的重要性。

本书可作为高等院校计算机及相关专业的“数据结构”教材,也可供其他相关人员阅读参考。

由于作者水平有限,本书可能存在很多不足,敬请读者批评指正。

作 者

2009年3月

目　录

第二部分 树形结构

第三部分 集 合

第五部分 算法设计基础

第1章 引 言

1.1 算法与数据结构

在学习数据结构之前，各位读者想必已学过了程序设计。在程序设计的学习中，编写解决某一个问题的程序时，必须先考虑如何存储待处理的数据以及根据数据处理的要求设计数据处理的算法。作为一门研究信息表示和处理的科学，计算学科设置了一门核心课程——数据结构，专门研究信息的存储和处理方法。

随着计算学科的发展，计算机的应用已经深入到各个领域。计算机处理的数据已不再局限于整型、实型等数值型数据，还可以是字符、表格、声音、图像等非数值型数据。数值计算的特点是数据类型简单、算法复杂，所以更侧重于程序设计的技巧。而非数值计算的特点是数据之间的关系复杂，数据量十分庞大，要设计出好的非数值计算的程序必须解决下列问题：

（1）明确所处理的数据之间的逻辑关系以及处理要求。非数值计算一般处理的是一批同类数据，如家谱管理系统处理的是家族中所有人员的信息，所需的操作有查找某个人的父亲，或查找某个人的孩子和所有的子孙；再如仓库管理系统处理的是一批库存信息，所需的操作包括改变某一产品的库存或查找某一产品的库存。因此，数据之间的逻辑关系包括两个层次：每个数据元素的组成，以及数据元素之间的关系。例如，家谱管理系统中，每个数据元素是家族中某个人的信息。描述一个人必须包含姓名、性别、出生年月等信息。数据元素之间的关系主要就是父子关系，其他关系都可以从父子关系中推导出来。

（2）如何将数据存储在计算机中。保存数据也要保存两个方面的内容：数据元素的保存以及数据元素之间的关系的保存。非数值计算中的数据元素很少是简单类型，一般都由多个部分组成，因此每个数据元素可以用程序设计语言中的记录型变量或对象来表示，而如何保存数据元素之间的关系则是数据结构研究的内容。

（3）如何实现数据的处理。数据元素之间的关系有各种保存方法，对每一种保存方法，数据处理的过程是不同的。每个数据处理的过程就是一个算法。

应用系统可以千变万化，但数据元素之间的逻辑关系的种类是有限的，数据结构抛开了各种具体的应用、具体的数据元素的内容，通过抽象的方法研究被处理的数据元素之间有哪些逻辑关系（称为逻辑结构），对于每种逻辑关系可能有哪些操作。然后研究每种逻辑关系在计算机内部如何表示（称为物理结构），对于每一种物理结构，对应的操作如何实现。每个数据结构处理一类逻辑关系，包括逻辑关系的物理表示以及运算的实现。因此，数据结构

的讨论可以分为两个层次:抽象层和实现层。抽象层讨论数据的逻辑结构和所需的运算。实现层讨论数据的存储表示及运算的实现。

掌握了数据结构以后,当要解决一个问题时,首先分析被处理的数据元素之间是什么关系,它需要完成哪些操作,然后选择一个合适的数据结构来处理数据,而不用针对每个应用来设计解决方法。

1.1.1 数据的逻辑结构

从概念上讲,一个数据结构是由一组同类的数据元素依据某种联系组织起来的。数据元素间的逻辑关系的描述称为数据的**逻辑结构**。不管应用如何变化,从抽象层面上看,数据的逻辑结构有下列4种。

(1) 集合结构:元素间的次序是任意的。元素之间除了"属于同一集合"的联系外没有其他的关系。例如,某次聚会中的所有人员,公共汽车上的所有乘客,存放在仓库中的产品。由于集合结构的元素间没有固有的关系,因此往往需要借助于其他结构才能在计算机中表示集合结构。

(2) 线性结构:数据元素之间构成一个有序序列。其中,第一个元素只有后继没有前驱,最后一个元素只有前驱没有后继。除了第一个和最后一个元素外,其余元素都有一个前驱和一个后继。例如,水泊梁山上的108条好汉,他们形成了一个集合,但他们之间有一个次序,宋江排第一,卢俊义排第二……

(3) 树形结构:除了一个特殊的根元素外,每个元素有且仅有一个前驱,后继数目不限。根元素没有前驱。树形结构表示的是一种层次关系。例如,一个家族就可以表示为树形结构。这家的老祖宗就是树根,老祖宗的儿子就是老祖宗的后继。每个人可以有多个儿子,因此后继数目不限。但每个人只能有一个父亲,因此只有一个前驱。老祖宗的儿子们又可以有他们的儿子,这样就形成了一棵树。老祖宗的父亲无历史记载,因此老祖宗就没有前驱。

(4) 图形结构:图是最一般的逻辑结构,图中的每个元素的前驱和后继数目都不限。例如,在一个计算机网络中,各个网络设备之间是由线路连接起来的。如果把连接看成是网络设备之间的关系,那么一个计算机网络的拓扑结构就形成了一个图状结构,因为每台网络设备都可以和多台其他设备相连接,向多台设备发送信息,也可以接收多台设备发来的信息。

以上4种结构如图1-1所示。有时也把线性结构以外的其他3种结构称为非线性结构。

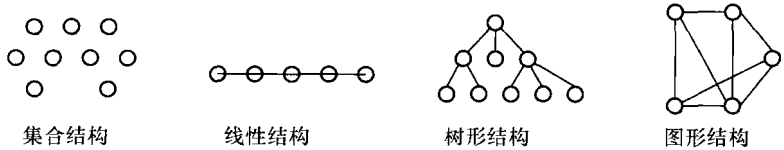

图1-1 数据的逻辑结构

一些表面上很不相同的数据也可以有相同的逻辑结构。例如,一个计算机网络可以用一个图形结构来表示,每个数据元素是一个网络设备,数据元素之间的关系是设备间的物理介质。如果两台设备间由一条物理媒体连接起来,则两台设备间有关系。如果用先导课程和后继课程作为课程之间的关系,则一个课程管理系统中的课程也是一个图形结构。一门课程可以有若干门先导课程,也可以有若干门后继课程。因此,逻辑结构是数据组织的本质。只要解决了这个本质问题,就可以把解决方案用到各种应用中去,而不必分别对每个问题进行研究。

关于逻辑结构,有以下几点需要特别说明:

(1) 逻辑结构与数据元素本身的内容无关。例如,在家谱管理系统中,不管如何描述一个人,人和人之间的关系总是树形的关系。当用姓名、性别、出生年月描述一个人时,是树形关系,如果每个人再加上一张照片后,还是树形关系。

(2) 逻辑结构与数据元素的个数无关。例如,在家谱中增加或删除某些人,整个家谱还是一个树形结构。

1.1.2 数据结构的运算

每种逻辑结构都有一定的处理要求,这些处理要求被称为数据结构的操作或运算。数据结构最常见的运算有如下几种。

- 创建运算(create):创建一个空的数据结构。
- 清除运算(clear):删除数据结构中的所有数据元素。
- 插入运算(insert):在数据结构指定的位置上插入一个新数据元素。
- 删除运算(remove):将数据结构中的某个数据元素删去。
- 搜索运算(search):在数据结构中搜索满足特定条件的数据元素。
- 更新运算(update):修改数据结构中的某个数据元素的值。
- 访问运算(visit):访问数据结构中的某个数据元素。
- 遍历运算(traverse):按照某种次序访问数据结构中的每一个数据元素,使每个数据元素恰好被访问一次。

除了上述运算之外,每种数据结构还可以包含一些特定的运算。例如,树形结构中,需要找某个数据元素的儿子,或找某个数据元素的父亲的操作。线性结构中,需要找某个元素的直接前驱或直接后继的操作。

数据元素之间的逻辑关系和数据结构的运算是数据结构不可分割的两个方面。一个数据结构就是针对某一个逻辑结构讨论数据的存储以及运算的实现,通常被称为**存储实现**和**运算实现**。

1.2 存储实现

存储实现的基本目标是建立数据的机内表示,它包括两个部分:数据元素的存储和数据元素之间的关系的存储。有时为了方便运算的实现,还可能会增加一些辅助信息的存储。数据的机内表示称为数据的物理结构。按照上述思路,一个物理结构由以下3个部分组成:

(1) 存储结点,每个存储结点存放一个数据元素。

(2) 数据元素之间的关系的存储,也就是逻辑结构的机内表示。

(3) 附加信息,为方便运算的实现而设置的一些"哑结点",如链表中的头结点。

其中,前两个部分是每个存储实现都必须具备的,第三部分则可根据运算实现的需要来决定是否设置。由于数据元素本身比较简单,因此存储结点的组织也比较简单,可能是一个基本的数据类型,也可能是一个记录类型或用户定义的类类型。因此,物理结构主要讨论的是数据元素之间的关系的表示。由于每个数据元素被表示为一个存储结点,所以逻辑结构就由存储结点之间的关联方式间接地表示。通常存储结点之间的关联有以下4种实现方式。

(1) 顺序实现:所有的存储结点存放在一块连续的存储区域中,结点之间的逻辑关系可以通过结点的存储位置来体现。在高级语言中,一块连续的存储空间通常可以用一个数组来表示。因此,顺序实现通常是用一个数据元素类型的数组来存储数据。

(2) 链接实现:存储结点可以分散地存放在存储器的不同位置,结点之间的关系通过一个指针显式地指出。因此,在链接存储中,每个存储结点包含两个部分:数据元素部分和指针部分。数据元素部分保存数据元素的值,指针部分保存一组指针,每个指针指向一个与本结点有逻辑关系的结点。

(3) 散列存储方式:是专用于集合结构的数据存储方式。在散列存储中,各个结点均匀地分布在一块连续的存储区域中,用一个散列函数将数据元素和存储位置关联起来。

(4) 索引存储方式:所有的存储结点按照生成的次序连续存放。另外设置一个索引区域表示结点之间的关系。

这些基本的存储方式以及它们的组合可以实现数据的灵活存储。

1.3 算法分析

一般而言,对同一个问题可以设计出不同的解决方法。因此,对于一个运算也可以有不同的实现,即可以设计出实现该运算的不同算法。这样就产生了如何评价这些算法的问题。通过这种评价,设计人员可以区分不同实现的相对的优劣,从而选择一个较好的算法。通常可以从以下几个方面来评价算法的质量。

(1) 正确性:算法应能正确地实现预定的功能。

(2) 易读性:算法应易于阅读和理解,以便于调试、修改和扩充。

(3) 健壮性:当环境发生变化(如遇到非法输入)时,算法能适当地做出反应或进行处理,不会产生不正确的运算结果。

(4) 高效率:具有较高的时间和空间性能。

这些指标往往是互相冲突的,例如,易读的算法一般效率较差。因此在实际评价中应根据需要有所侧重。数据结构课程着重讨论算法的时空性能,这并不意味着这一指标比其他指标更重要,而仅仅是由于学习阶段所限。

确定算法的时空性能通常被称为**算法分析**。算法的时空性能是指算法的**时间性能**(或称**时间复杂度**)和**空间性能**(或称**空间复杂度**)。前者指算法包含的计算量,后者指算法需要的存储量。

空间复杂度的讨论比较简单,它关心的是除了被处理的数据所占空间之外所需的额外空间的大小。空间复杂度一般按最坏情况来分析。下面主要讨论时间复杂度的分析。

1.3.1 时间复杂度的概念

算法的具体实现是程序。程序的运行时间是程序运行从开始到结束所需要的时间。影响程序运行时间的因素很多,主要有以下几个:

(1) 问题规模和输入数据的分布。

(2) 编译器生成的目标代码的质量。

(3) 计算机系统的性能。

(4) 程序采用的算法的优劣。

当在不同的计算机上运行同一个程序,输入同样的数据,所需的运行时间是不同的。这是因为计算机的硬件性能可能不同,如 CPU 的速度、内存的大小。即使硬件环境一样,但运行的软件环境不一样,运行时间也可能不一样。因为不同的编译器生成的目标代码的质量是不一样的,还有操作系统运行程序的过程也会影响程序运行的时间。这就是说,程序运行所需要的时间取决于计算机软、硬件系统。

但如果排除计算机的因素,假定在相同的环境下运行解决相同问题的程序,情况又会如何呢? 很显然,对于求解同一问题的不同算法,其程序运行的时间是不同的。好的算法需要较少的时间。算法自身的好坏是影响运行时间的根本因素。例如,用不同的排序算法对同一组数据进行排序,程序运行的时间是不同的。因此,程序的运行时间也取决于算法的设计。

同样的一个算法,如果处理的数据量不一样,运行时间也不一样。例如,执行一个排序算法排序 100 个元素和排序 1 000 个元素所需的时间是不同的。因此,运行时间也是处理的数据量(也称为问题规模)的函数。

如果使用同一个程序,问题的规模也相同,在同一个软、硬件环境下的运行时间是否一定相同呢? 也不一定。同样排序 100 个数据,如果这 100 个数据本身是有序的或这 100 个