面向21世纪应用型本科计算机规划教材

数据结构

(C语言版)

杨 升 主编

厦门大学出版社
XIAMEN UNIVERSITY PRESS

数据结构

(C语言版)

主　编　杨　升
副主编　林　芳　肖钟捷
参　编　丁　潇　潘俊虹

厦门大学出版社

图书在版编目(CIP)数据

数据结构:C语言版/杨升主编. —厦门:厦门大学出版社,2009.8
(面向21世纪应用型本科计算机规划教材)
ISBN 978-7-5615-3255-3

Ⅰ.数… Ⅱ.杨… 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.TP311.12 TP312

中国版本图书馆CIP数据核字(2009)第114335号

厦门大学出版社出版发行
(地址:厦门市软件园二期望海路39号 邮编:361008)
http://www.xmupress.com
xmup@public.xm.fj.cn
厦门市明亮彩印有限公司印刷
2009年8月第1版 2009年8月第1次印刷
开本:787×1092 1/16 印张:15.25
字数:371千字 印数:1-3 000册
定价:24.00元

前 言

“数据结构”是计算机学科的专业核心基础课程，是一门集理论性和实践性于一体的课程。

“数据结构”的教学目的是：首先，让学生理解什么是“数据结构＋算法＝程序”，即让学生懂得设计程序求解问题时，不仅仅要实现一个有效、合理的算法，还要求设计出与之结合的恰当的数据结构；其次，培养学生的抽象能力，即如何为应用中所涉及的数据选择适当的逻辑结构、存储结构及相应算法；最后，培养学生的实践编程能力，使之编写的程序符合软件工程规范。总之，通过本门课程的学习，为学生学习后续课程和将来进行软件开发等打下坚实的基础。

本教材具有以下特点：

- 每章开头的“知识点”和“学习要求”对本章的学习起到了“提纲挈领”的作用。
- 全书对于理论知识点的讲解，循序渐进，思路清晰；针对每一个知识点，都配有相应的实例说明。
- 所有算法（数据操作）都用C语言函数实现，几乎不用做任何修改就可被其他函数调用。为方便初学者实践验证“数据结构”的算法，前面几章还给出了算法实现的完整C源程序。
- 基本上每章都配有实验及实验指导和习题，以检验每章的学习效果。

在计算机科学与技术领域，数据结构作为一门学科，主要研究数据的各种逻辑结构和存储结构，以及对数据的各种操作。所以，“数据结构”教材主要包括三个方面的内容：数据的逻辑结构、数据的物理结构、对数据的操作（也称为算法，其设计取决于数据的逻辑结构，而实现取决于数据的物理存储结构）。

本书共分9章：第1章概述了数据结构的一些基本概念和术语，以及算法和算法分析的概念；第2章介绍线性表的逻辑结构和存储结构及运算实现；第3章介绍特殊线性表实例——栈和队列的定义及其存储结构，以及应用实例；第4章介绍串的基本概念及基本操作应用；第5章介绍数组和广义表的定义及存储结构；第6章主要介绍树和二叉树的定义及基本术语，详细讨论了二叉树的遍历运算及其应用；第7章介绍图的基本概念和图的几种存储结构，特别介绍了图的遍历、生成树、拓扑排序、关键路径、最短路径等内容；第8章介绍了查找，包括静态查找表、动态查找表和哈希表；第9章主要讨论了各种内部排序算法及其性能分析。

参与本教材编写的都是长期在一线从事“数据结构”课程教学的老师。其中，第1章由杨升编写，第2、3章由丁潇编写，第4、5章由潘俊虹编写，第6、7章由肖钟捷编写，第8、9章由林芳编写。全书由杨升、肖钟捷统稿。

在本教材的编写过程中，感谢厦门大学出版社的大力支持。感谢所有关心本书出版，为本书出版付出辛劳的人们！感谢从事“数据结构”教学和研究的前辈们，没有他们的积累，就没有

我们今天的成功。

本书是针对“应用型本科”计算机类专业学生编写的，同时也适合作为计算机类专科和其他信息类相关专业本、专科的教材。

由于编者水平有限，加之时间仓促，错误和不足之处在所难免，敬请读者批评指正，不吝赐教。

本教材配有相应的电子教案及每章后面的实验和习题解答。如有需要，可通过邮箱npxzj66@sohu.com联系。

编　者

2009年6月

目 录

第 1 章 概　论

本章知识点：

- 什么是数据结构
- 数据结构的重要性
- 基本概念和基本术语
- 什么是算法
- 算法性能的度量

本章学习要求：

(1) 了解数据结构在计算机学科中的地位和重要性。

(2) 掌握数据结构的概念及其与数据类型的区别。

(3) 掌握算法的概念和算法的复杂度分析方法。

1.1　什么是数据结构

信息是人们对现实世界中各种事物的描述，计算机学科是一门研究如何用计算机对信息进行表示和处理的科学。信息在计算机中的表示及组织结构又直接影响甚至决定了计算机对信息的处理。随着计算机应用技术的发展，需要计算机表示和处理各种各样的多媒体数据，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，必须研究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程所要研究的问题。通过学习它，也会为计算机专业后续课程（如软件工程、编译原理和操作系统等）的学习打下坚实的基础。

1.1.1　基本概念和术语

对现实世界中的事物及活动，人们利用文字、数字以及其他规定的符号所做的抽象描述，称为信息。从计算机的角度来看，所有被输入到计算机中，被计算机处理的符号化了的信息，我们称之为数据。可见数据是信息在计算机中的载体（或者说表示形式），它能够被计算机识别、存储和加工处理。计算机内的数据可包含整数、实数、字符串、图像和声音等各种类型。归类起来，我们可以得到以下几个有关数据的基本概念：

数据(Data)：是对信息的一种符号表示。在计算机科学中是指所有能输入到计算机中并被

计算机程序处理的符号的总称。

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。在有些情况下,数据元素也称为元素、结点、顶点、记录等。一个数据元素可由若干个数据项组成。**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 $N=\{0,\pm1,\pm2,\pm3,\cdots\}$,但计算机中的整数数据对象集合 $N1$ 应该是上述集合 N 的一个子集,$N1=\{0,\pm1,\pm2,\cdots,\pm\text{maxint}\}$,其中 maxint 是依赖于所使用的计算机和语言的最大整数。

数据结构(Data Structure):是相互之间存在一种或多种特定关系的数据元素的集合。因此,可以把数据结构看成是带结构的数据元素的集合,其一般包括以下三个方面内容:

(1) 数据元素之间的逻辑关系,即数据的逻辑结构。

数据的逻辑结构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。

(2) 数据元素及其关系在计算机存储器中的存储实现,即数据的存储结构,也称为数据的物理结构。

数据的存储结构是逻辑结构用计算机语言的实现(亦称为映象),也就是逻辑结构在计算机中的存储方式,它依赖于计算机语言。一般只在高级语言(如 C 语言)的层次上来讨论存储结构。

(3) 施加在该数据上的操作,即数据的运算。

数据的运算是定义在数据的逻辑结构之上的,每种逻辑结构都有一组相应的运算。例如,最常用的运算有检索、插入、删除、排序等。数据的运算最终需在对应的存储结构上用算法实现。

如上所述,对于数据元素互相之间的逻辑关系,我们称为数据的逻辑结构。在不会产生混淆的前提下,常常将数据的逻辑结构简称为数据结构。数据的逻辑结构主要有以下 4 类(参见图 1-1):

(a) **集合**:其中的数据元素之间除了"属于同一个集合"的关系以外,别无其他关系。

(b) **线性结构**:其中的数据元素之间存在一对一的(线性)关系。

(c) **树型结构**:其中的数据元素之间存在一对多的(层次)关系。

(d) **图状结构**(或称**网状结构**):其中的数据元素之间存在多对多的关系。如因特网的网页链接关系就是一个非常复杂的图结构。

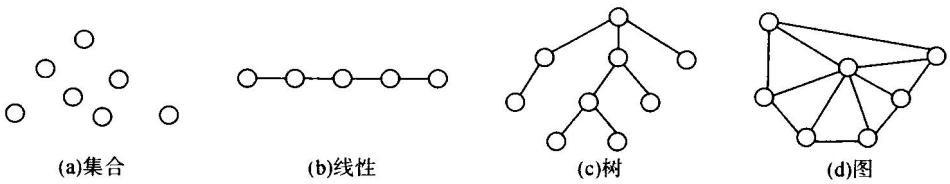


图 1-1 4 类基本结构关系图

下面举一个简单的数据结构例子。

例 1-1　如表 1-1 所示，某市电话号码表是一个数据结构。

表 1-1　某市电话号码表

姓　名	移动电话号码	家庭电话号码
张静文	13812345678	51881666
萧　晗	13912345678	51661677
陈连江	13712345678	51991688
张　华	15812345678	51991699
陈金瑜	15912345678	51861611
⋮	⋮	⋮

表 1-1 称为一个数据结构。表中的每一行（记录）是一个结点（数据元素）。每个数据元素由 3 个数据项（即姓名、移动电话号码和家庭电话号码）组成。

从表 1-1 中还可以看出，该表中的记录（结点）之间的逻辑关系具有的特点是：开始结点和终端结点都是唯一的，除了开始结点和终端结点以外，其余结点都有且仅有一个前驱，有且仅有一个后继（一对一关系）。显然，表 1-1 是一种典型的线性结构。

1.1.2　数据的存储结构

数据结构在计算机中的存储表示（又称映射）称为数据的物理结构，即数据的存储结构。其包含数据元素的表示和关系的表示。实际上，数据的存储结构是建立一种由逻辑结构到存储空间的映射。

下面将分别介绍 4 种基本的存储映射方法。

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里，结点之间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储表示称为顺序存储结构（Sequential Storage Structure），通常顺序存储结构是借助于计算机程序设计语言（例如 C 语言）的数组来描述的。

顺序存储主要应用于线性数据结构的物理存储。对于非线性的数据结构，则要通过某种线性化的方法来实现其顺序存储。

顺序存储方法也称为紧凑存储结构，其主要优点是节省存储空间，因为分配给数据的存储单元全用于存放结点的数据，结点之间的逻辑关系（用存储空间的邻接关系表示）不需占用存储空间。另外，顺序存储方法可实现对结点的随机存取。即每个结点按逻辑顺序对应有一个序号，由该序号可直接计算出结点的存储地址（例如在 C 语言中，对于数组 A，其序号为数组元素的下标，数组元素 A[i] 的物理存储地址即为 A + i）。顺序存储方法的主要缺点是不便于修改，在对结点的插入、删除运算时，可能要移动一系列结点。

2. 链式存储方法

该方法不要求逻辑上相邻的结点在物理位置上也相邻，结点间的逻辑关系是由附加的指

针字段表示的。由此得到的存储表示称为链式存储结构,通常要借助于计算机程序设计语言的指针类型来描述。

链式存储方法的主要优点是便于修改,在进行插入、删除运算时,仅需修改相应结点的指针域,不必移动结点。但与顺序存储方法相比,链式存储方法的主要缺点是存储空间的利用率较低,因为分配的存储单元有一部分被用来存储结点之间的逻辑关系了。所以也可以说链式存储方法是一种用空间换取时间的方法。另外,由于逻辑上相邻的结点在存储空间中不一定相邻,所以不能对结点进行随机存取。

3. 索引存储方法

索引法是顺序存储法的一种推广。该方法通常是在存储结点信息的同时,还建立附加的索引表。索引表中的每一项称为索引项,若每个结点在索引表中都有一个索引项,则该索引表称为稠密索引(Dense Index);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(Spare Index),参见图 1-2。索引项的一般形式是:(关键字,地址),关键字是能唯一标识一个结点的那些数据项。稠密索引中索引项的地址指示结点所在的存储位置,稀疏索引中索引项的地址指示一组结点的起始存储位置。

这种带有索引表的存储结构可以大大提高数据查找的速度,其缺点是增加了一个索引表,降低了存储空间的利用率。

4. 哈希(或散列)存储方法

该方法的基本思想是根据结点的关键字通过哈希(或散列)函数直接计算出一个值,并将该值作为该结点的存储地址。

哈希存储方法的优点是查找(称为哈希查找)速度快,只要给出待查结点的关键字,就可立即计算出该结点的存储地址。但与前三种存储方法不同的是,哈希存储方法不存储结点之间的逻辑关系,所以哈希存储方法一般只适用于要求对数据能够进行快速查找和插入的场合。

上述 4 种基本的存储方法,既可以单独使用,也可以组合起来对数据结构进行存储映射。同一种逻辑结构采用不同的存储方法,可以得到不同的存储结构。选择何种存储结构来表示相应的逻辑结构,要视具体要求而定,主要考虑的是运算方便及算法的时空要求。

1.1.3 数据结构与数据类型

数据结构不同于数据类型,也不同于数据对象,它不仅要描述数据类型的数据对象,而且要描述数据对象各元素之间的相互关系。

数据结构的形式定义为:数据结构是一个二元组

$$\text{Data} - \text{Structure} = (D, S)$$

其中:D 是数据元素的有限集,S 是 D 上关系的有限集。举例说明如下:

例 1-2 复数的数据结构定义如下:

$$\text{Complex} = (C, R)$$

其中:C 是含两个实数的集合$\{C1, C2\}$,分别表示复数的实部和虚部。这里关系集 $R = \{P\}$,P 是定义在集合上的一种关系$\{\langle C1, C2\rangle\}$,有序偶$\langle C1, C2\rangle$ 中的 $C1$ 是复数的实部,$C2$ 是复数的

虚部。

存储结构是数据结构不可缺少的一个方面：同一逻辑结构的不同存储结构可冠以不同的数据结构名称来标识。

例如线性表是一种逻辑结构，若采用顺序存储方法的存储表示，称为顺序表；若采用链式存储方法，称为链表；若采用哈希存储方法，则可称为哈希表。

数据的运算也是数据结构不可分割的一个方面。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性质的不同，也可能导致完全不同的数据结构。

例如对线性表（顺序表或链表）限定插入、删除操作只在表的一端进行，则该线性表称为栈；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，则该线性表称为队列。

所谓**数据类型**是一个值的集合以及在这些值上定义的一组操作的总称。例如 C 语言中的整型数据，其值为某个区间内的整数（和机器有关），定义在其上的操作为加、减、乘、除等算术运算。

数据类型可分为原子类型（如 C 语言中的实型、字符型等）和结构类型（如 C 语言中的数组、结构体等）两种。

通常数据类型可以看作是程序设计语言中已实现的数据结构。数据类型也可以定义为一种数据结构和对该数据结构允许进行的运算集。对于原子类型，其数据结构就是相应取值范围内的所有数据，每一个数据值是不可分的独立整体，因而数据值内部无结构可言。对于结构类型，其数据结构就是相应元素的集合（该结构类型中的一个值）和元素之间所含关系的集合。

总之，数据结构是指计算机处理的数据元素及其组织形式和相互关系，而数据类型是某种程序设计语言中已实现的数据结构。利用程序设计语言提供已有数据类型，可以根据从问题中抽象出来的各种数据模型，构造出描述这些数据模型的各种新的数据结构。

1.2　为什么要学习数据结构

1.2.1　数据结构的重要性

“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（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。比如在研究信息检索时，必须考虑到如何组织存储数据，以便在查找和存取数据元素时更加有效；又比如在嵌入式系统应用中，由于内存的因素，数据的存储结构、搜索时间以及空间开销都得考虑。所以数据结构显得尤其重要。其实，在计算机科学中，“数据结构”不仅是一般程序（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、操作系统及其他大型数据库系统程序的重要基础。

随着计算机应用领域的不断扩大，非数值计算问题占据了当今计算机应用的绝大部分，各

数据元素之间的复杂联系已经不是普通数学方程式所能表达的了,无论设计系统软件还是应用软件都会用到各种复杂的数据结构。因此,掌握好数据结构课程的知识,对于提高解决实际问题的能力将会有很大的帮助。著名的计算机科学家沃思(N. Wirth)指出:算法 + 数据结构 = 程序。这里的数据结构包含数据的逻辑结构和存储结构。可见,一个"好"的程序要选择一个合理的数据结构和好的算法,而好的算法的选择很大程度上取决于描述实际问题所采用的数据结构,所以,要想编写出"好"的程序,仅仅学习计算机语言是不够的,必须扎实地掌握数据结构的基本知识和基本技能。

软件质量好坏有一个很重要的因素:软件体系结构。软件体系结构就好比房子的结构,软件体系结构的规划设计也就决定了软件本身的结构(软件的环境,功能模块的安排,位置,互相联系关系等),而拥有良好数据结构和算法的功能函数或函数块也是整个软件质量的根本保证。就好比只有拥有质量过硬的砖头和横梁及合理的框架结构,房子的质量和寿命才能得到根本保证。一样道理,在从事软件开发时,只有我们拥有了足够好的、经过优化的功能函数或功能模块,严格按照软件工程规范,合理运用面向对象方法,按照科学的体系结构要求,才能在经济的条件下,开发出优良(准确,健壮,时空效率高,易修改补充,美观,易使用)的软件。

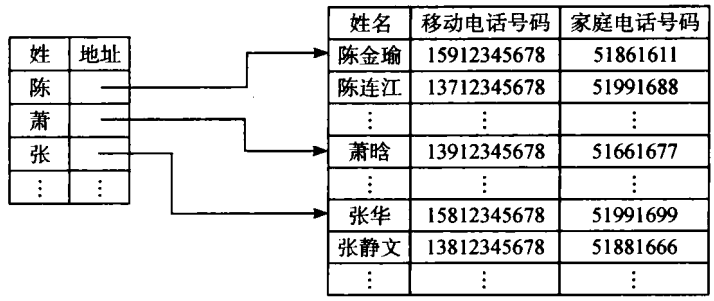
1.2.2 数据结构的一个应用例子

通过上面的讨论,我们知道,在利用计算机解决现实生活中的问题时,要针对实际问题选择一种好的数据结构,加之设计一个好的算法,而好的算法在很大程度上取决于描述实际问题的数据结构。

比如要编写一个查询某个城市或单位的私人电话号码的程序,程序功能要求对任意给出的一个姓名,若该人有电话号码,则迅速找到其电话号码信息,否则指出该人没有电话号码。

要解决此问题首先要构造一张电话号码登记表,如表 1-1 所示的某市电话号码表。表中每个结点存放数据项信息必须包括姓名和电话号码。根据所给出的姓名信息,可以在该表中查到该姓名的相应电话信息。

但是能否高效地实现电话号码的检索功能,还取决于这张表的结构及存储方式。表 1-1 最简单的存储方式是将表中结点顺序地存储在计算机中。当查找某人的电话号码时从头开始依次查对姓名,直到找出正确的姓名或是找遍整个表均没有找到为止。这种查找算法对于一个不长的电话号码表或许是可行的,但对一个有百万以上私人电话的城市就不实用了。若这张表是按姓氏排列的,则可另外构造一张姓氏索引表,采用如图 1-2 所示的存储结构。那么查找过程是:先根据姓名在姓氏索引表中查对姓氏,然后根据索引表中查到姓氏对应的地址到电话号码登记表中核查姓名,这样查找登记表时就无需查找其他姓氏的名字了。显然,在这种新的结构上产生的查找算法就更有效。通过对比表 1-1 和图 1-2 可知,实现高效的电话号码表检索功能是建立在有序(或部分有序)的电话号码表和附加的姓氏索引表基础上的。



(a)姓氏索引表　　(b)按姓名排序的电话号码表

图 1-2　采用索引存储的电话号码表

1.3　算法和算法分析

1.3.1　算法的特点

1. 什么是算法

算法(Algorithm)是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。它以零个或多个值作为输入，并产生一个或多个值作为输出。

一个算法可以用自然语言、计算机程序语言或其他语言来描述，唯一的要求是该说明必须精确地描述计算过程。

一般而言，描述算法最合适的语言是介于自然语言和程序语言之间的伪语言。它的控制结构往往类似于 Pascal、C 等程序语言，但其中可使用任何表达能力强的方法使算法表达更加清晰和简洁，而不至于陷入具体的程序语言的某些细节。从易于上机验证算法和提高实际程序设计能力的角度考虑，本书采用 C 程序设计语言来描述算法。

算法具有以下五个明显的特性：

(1) 有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都在有穷时间内完成。

(2) 确定性：算法中每一条指令必须有确切的含义，不存在二义性，且算法只有一个入口和一个出口。

(3) 可行性：一个算法是可行的，即算法描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

(4) 输入：一个算法有零个或多个输入，这些输入取自于某个特定的对象集合。

(5) 输出：一个算法有一个或多个输出，这些输出是同输入有着某些特定关系的量。

2. 算法设计的要求

一个好的算法设计要达到以下几个目标：

(1) 正确性(Correctness)：算法应满足具体问题的需求。