国外计算机科学经典教材

Data Structures Using C And C++

Second Edition

数据结构

C和C++语言描述

(第2版)

Yedidyah Langsam
(美) Moshe J. Augenstein 著
Aaron M. Tenenbaum
李 化 萧 东 译

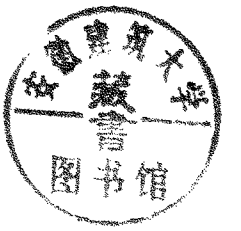
清华大学出版社

国外计算机科学经典教材

数据结构
C 和 C++语言描述
(第 2 版)

Yedidyah Langsam
(美) Moshe J. Augenstein 著
Aaron M. Tenenbaum

李 化 潇 东 译



清华大学出版社

北 京

EISBN: 0-13-036997-7

北京市版权局著作权合同登记号 图字：01-2003-3092

图书在版编目(CIP)数据

数据结构 C 和 C++语言描述/(美) 兰山姆(Langsam,Y.)，(美)奥根斯坦(Augenstein,M.J.)，(美)特内巴姆(Tenenbaum,A.M.) 著；李化 潇东译. —2 版. —北京：清华大学出版社，2004

书名原文：Data Structures Using C And C++, Second Edition

(国外计算机科学经典教材)

ISBN 7-302-08068-2

Ⅰ. 数… Ⅱ.①兰… ②奥… ③特…④李…⑤潇… Ⅲ.①数据结构 ②C 语言—程序设计

Ⅳ.①TP311.12②TP312

中国版本图书馆 CIP 数据核字(2004)第 008224 号

出 版 者：清华大学出版社 **地 址**：北京清华大学学研大厦
http://www.tup.com.cn **邮 编**：100084
社 总 机：010-62770175 **客户服务**：010-62776969
组稿编辑：曹康
文稿编辑：于平
封面设计：康博
版式设计：康博
印 刷 者：北京市清华园胶印厂
装 订 者：三河市金元装订厂
发 行 者：新华书店总店北京发行所
开 本：185×260 **印张**：35.25 **字数**：902 千字
版 次：2004 年 3 月第 1 版 2004 年 3 月第 1 次印刷
书 号：ISBN 7-302-08068-2/TP·5837
印 数：1～4000
定 价：68.00 元

前　言

这是一本为两个学期的数据结构和编程的课程而设计的教材。曾经有几年，我们把数据结构课程的教学对象定为那些已经学过一个学期的高级语言编程和一个学期的汇编语言编程的学生。我们发现编程技巧的教学花费了相当一部分时间，因为这些学生尚未充分适应编程，并且还不能独立实现抽象结构。天资高一些的学生最终能够理解程序所做的事情；而差一些的学生从来都没能理解。基于这些经验，我们已经确信要先学习数据结构课程，并且必须与随后的编程课程紧密结合起来。这本教材就是这一思想的产物。

本教材介绍了抽象概念，演示了这些概念怎样在解决问题中发挥作用，并且通过一门编程语言演示怎样使它们具体化。书中对于概念的抽象化和具体化作了同等的强调，这样学生可以学到概念本身、概念的实现及其应用。

本教材中使用的编程语言是 C 和 C++。C 非常适合于这门课程，因为它包括了编写可读的程序所必需的控制结构，并允许使用多种不同的方法实现诸如堆栈、链表、树等基本数据结构。这使学生能够意识到在实际编程中程序员需要面对的选择和权衡。C 语言广泛应用到许多不同的计算机，并且其普及范围仍在扩大。正如 Kernighan 和 Ritchie 所指出的，C 是“一门令人愉快的、富于表达能力的、通用的语言”。

我们在前几章中加入了 C++的知识，介绍了 C++的特点及其在实现数据结构中的用法。对于 C++不需要特别的背景知识。在新增加的 1.4 节中，我们介绍了 C++中的类，以及类的成员函数。其中还介绍了继承和面向对象的概念。该节包括一个用 C++实现抽象数据类型以及多态性的例子。在 2.3 节我们增加了使用 C++模板的堆栈的实现。我们通过它演示怎样为不同的基本类型对复杂数据结构进行参数化。新增加的 4.6 节演示了怎样用 C++实现链表，它表明了封装在实现数据结构中的局限性和能力。其中的要点是我们必须小心地封装数据结构，以允许用户能够对它执行所需的操作。另外我们在其中还讨论了 C++中内存的动态分配和释放。

对于使用本教材的学生，惟一需要预备的知识是一个学期的编程语言课程的学习。已经学过诸如 FORTRAN、Pascal、PL/I 等编程语言课程的学生可以将本教材和基础 C 或 C++教材一起使用。第 1 章还提供了帮助这些学生熟悉 C 语言所需的知识。

第 1 章是数据结构的介绍。1.1 节介绍抽象数据结构的概念以及实现的概念。1.2 和 1.3 节介绍 C 语言中的数组和结构，并涵盖了这两种数据结构的实现和应用。第 2 章讨论堆栈及其 C 语言实现。由于这是介绍的第一个数据结构，因此详细讨论了在其实现中容易出现的错误。2.3 节介绍了后缀、前缀和中缀表示法。第 3 章讨论递归、递归的应用和实现。第 4 章介绍队列、优先队列、链表以及它们的实现。我们给出了两种实现方法，一种使用可用结点数组，另一种使用动态存储。第 5 章讨论树。第 6 章介绍大 O 表示法和排序。第 7 章讲述内部搜索和外部搜索。第 8 章介绍图。第 9 章讨论存储管理。

课时为一个学期的数据结构课程内容包括：1.1 节、第 2 章到第 7 章、8.1 节和 8.2 节，以

及 8.4 节的一部分。如果时间较紧，第 3、6、7、8 章的部分内容可以略去。

本书适合于作为如下课程的教材：基于算法和数据结构知识单元(AL 1-6, 8)，以及编程语言知识单元(PL3-6, 10, 11)的课程(见 ACM/IEEE-CS Joint Curriculum Task Force 的报告 Computing Curricula 1991)。书中的数据结构示例和算法过程分析与报告中给出的非常接近，本书可以作为计算机科学专业或非计算机科学专业的第二或第三级课程。

本教材适合于 Curriculum 78(Communications of the ACM, 1979 年 3 月)中的 C82 课程和部分 C87、C813 课程，信息系统本科教程(Communications of the ACM, 1973 年 12 月)中的 UC1 和 UC8 课程，以及 Curriculum 68 的 I1 课程(Communications of the ACM, 1968 年 3 月)。本书特别涵盖了 Curriculum 78 中的 P1、P2、P3、P4、P5、S2、D1、D2、D3 和 D6 中的部分或全部主题。

算法是用一种介于英语和 C 语言之间的中间语言描述的，在 C 语言风格的代码中穿插了一些英语。这些算法可以使读者把注意力集中在解决问题所使用的方法上，而无需关心变量的声明或实际语言的某些特别要求。这些问题我们将在把算法转换到程序时介绍，并将指出在转换过程中一些容易发生的错误。

程序和算法所使用的缩进格式基本上是按照 Kernighan 和 Ritchie 建议的“The C Programming Language, Prentice Hall, 1978”一书中提出的格式，同时我们发现它非常有用。我们还采用了在每个有(})的地方加注注释的习惯，以此注明当前结束的指令。这种习惯和缩进格式都是增加程序可读性的有效工具。

书中的大多数概念都通过若干个例子来阐释。这些例子中有的本身就是非常重要的论题(如前缀表示法、多字运算等)，可以对它们给予相应的重视。其他例子演示了不同的实现技术(如树的顺序存储)。教师可以根据自己的需要自由选择或多或少的例子进行讲解。这些例子也可以安排学生独立阅读。可以预见的是，教师无法在一个或两个学期的课程中把所有例子讲解得足够详细。我们认为，根据本书所预期的学生的水平，深入详细地讲解若干个例子比大范围地粗略讲解更加重要。

书中的所有程序和算法都已经进行了测试和调试。我们要感谢 Miriam Binder 和 Irene LaClaustra 为此作出的巨大帮助。他们对于工作的热诚远远超出了职责的要求，同时他们还提出许多有价值的建议。因此，书中仍然存在的错误完全是作者的责任。

书中提供了许多不同的类型和难度的练习题。有的是练习性的，以强化对知识点的理解；有的要求对书中的程序或算法进行修改；还有的引入了新概念，从而更具挑战性。通常，一组连续的练习题完整地提出了一个新的课题，可以作为一个学期的课程设计或附加讲座。教师应该慎重地布置练习题，以保证作业与学生的水平相适应。我们认为每个学期给学生布置一些编程项目(根据不同的难度可以有 5 到 12 个)是非常必要的。在练习题中就提供了一些这种类型的项目。

在使用 C 语言时，我们力图遵循 Kernighan 和 Ritchie 在第二版 C 语言教材中所述的内容。这也与 ANSI C 标准一致。书中给出的程序都是用 Borland C++开发的，但是只使用了 ANSI C++草案标准中的特性。这些程序应该可以不经修改地在各种 C++编译器上运行。当然，教师还应当提醒学生注意其所用编译器的一些特性。我们也增加了与若干种个人计算机上的 C 和 C++编译器有关的参考信息。

Miriam Binder 和 Irene LaClaustra 花了大量时间来输入并校正本书的原稿，他们还负责管理一大组学生。我们不断地对原稿进行整理，不时地增删一些内容，他们的合作和耐心尤其值得我们衷心的感谢。

我们要感谢 Shaindel Zundel-Margulis、Cynthia Richman、Gittie Rosenfeld-Wertenteil、Mindy Rosman-Schreiber、Nina Silverman、Helene Turry，以及 Devorah Sadowsky-Weinschneider，感谢他们的宝贵帮助。

这里我们要特别提到城市大学计算中心的员工们。他们用中心的卓越设施为我们提供了极大的帮助。同样的感谢还要献给 Brooklyn 大学计算中心的员工们。

我们要感谢 Prentice Hall 的编辑和员工们，特别是复审人员，因为他们提出了许多有价值的建议和意见。

最后，我们要感谢我们的妻子们，Vivienne Esther Langsam、Gail Augenstein 和 Miriam Tenenbaum，感谢她们在本书漫长而艰辛的编写过程中给我们的建议和鼓励。

Yedidyah Langsam
Moshe J. Augenstein
Aaron M. Tenenbaum

目　　录

第1章　数据结构入门

计算机是一种处理信息的机器。要学习计算机科学，就要学习如何在计算机中组织、处理信息，以及如何利用这些信息。因此，对于计算机科学专业的学生而言，理解信息的组织和处理对于他们进一步的学习是至关重要的。

1.1　信息和涵义

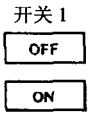
如果计算机科学本质上是对信息的研究，那么首先要提出的问题是：什么是信息？遗憾的是，尽管信息的概念是整个计算机科学领域的基石，我们却无法精确地回答这个问题。在这一点上，计算机科学中的信息与几何学中的点、线、面的概念非常相似：它们都是没有定义的术语，可以组成其他声明，但是无法用更基础的概念来解释。

在几何学中，我们可以讨论线的长度，虽然线这个概念本身是没有定义的。线的长度是一种量化度量。与此相似，在计算机科学中信息也有量化的度量。信息的基本单位是“比特”(bit)，它的值可以表示两种互斥的可能状态。例如，电灯开关可以处于两个位置中的一个，但是不能同时处于这两个位置，那么它的位置就或者是“开”，或者是“关”——这个事实就是一个比特的信息。如果一个设备可以有多于两个的状态，那么它处于某个特定状态的事实就是一个比特以上的信息。例如，假定一个拨号盘有 8 个可能的位置，那么它在 4 这个位置就排除了其他的 7 种可能；而电灯开关在“开”的位置仅排除了另一种可能状态。

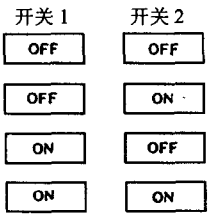
另一种考虑这个现象的思路如下：假设我们只有双路开关，但是可以根据需要使用多个这种开关。为了表达一个具有 8 个位置的拨号盘，我们需要多少个这样的开关呢？很明显，一个开关只能表示两种位置(见图 1-1a)。两个开关可以表示 4 种不同的位置(图 1-1b)，而 3 个开关可以表示 8 种不同的位置(图 1-1c)。一般来说，n 个开关可以表示 2^n 种不同的可能状态。

我们用二进制数字 0 和 1 来表达一个比特的两种可能的状态(事实上，bit 这个词是(binary digit(二进制数字)的缩写)。给定 n 个比特，一个由 n 个 1 或 0 组成的串表达了它们的设置。例如，串 101011 表示 6 个开关，第 1 个开关是“开”(1)，第 2 个是“关”(0)，第 3 个是开，第 4 个是关，第 5 个和第 6 个是开。

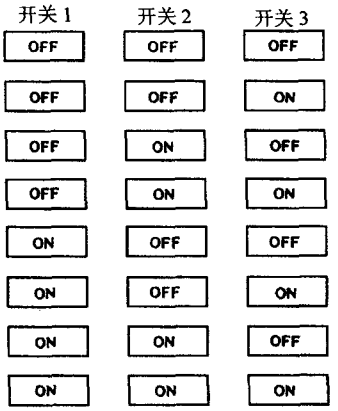
我们已经明白了 3 个比特就足以表示 8 种可能状态。这 3 个比特的 8 种可能配置(000，001，010，011，100，101，110，111)可以用来表示整数 0 到 7。然而，对于用哪个特定的设置表示哪个特定的整数并没有任何固有的规则。给比特设置赋任何一个整数值都是合法的，只要不是两个整数值赋给同一个比特设置。一旦完成了这样的赋值，一个特定的比特设置就可以明确地解释为一个特定的整数。让我们介绍一些广泛使用的把比特设置解释为整数的方法。



(a) 一个开关(两种可能)



(b) 两个开关(4 种可能)



(c) 3 个开关(8 种可能)

图 1-1　开关的可能状态

1.1.1　二进制整数和十进制整数

在把比特设置解释为非负整数的方法中，用得最为广泛的是二进制数系统(binary number system)。在这个系统中，每个比特位置表示 2 的一个幂。最右边的比特位置表示 2^0，等于 1；它左边相邻的一个比特位置表示 2^1，等于 2；再向左的一个比特位置表示 2^2，等于 4；等等。可以把整数表示为 2 的幂的和。一个全是 0 的串表示数字 0。如果一个 1 出现在某个特定的比特位置，那么该位置所代表的 2 的幂就被累加到和中；如果是 0，那么该位置所代表的 2 的幂就不被累加到和中。例如，比特组 00100110 在 1、2、5 位置上是 1(从右至左计算位置，最右边的为位置 0)。因此 00100110 代表整数 $2^1+2^2+2^5 = 2+4+32 = 38$。在该种解释下，任一串长度为 n 的比特表示 0 到 2^n-1 之间的惟一一个非负整数，而任一个介于 0 到 2^n-1 之间的非负整数也可以表示一个惟一的长度为 n 的比特串。

有两种常用的表示负二进制数的方法。第一种方法称为“1 的补码表示法”(ones complement notation)。在这种方法下，一个负数是通过把其绝对值的每一个比特取反来表示的。例如，00100110 表示 38，因此 11011001 就用来表示-38。这意味着数的最左边的比特不再用来表示一个 2 的幂，而被保留为这个数的(正负)符号。一个以 0 开始的比特串表示一个正数，而以 1 开