国外计算机科学教材系列

数据结构与算法分析

DATA STRUCTURES AND ALGORITHM ANALYSIS

CLIFFORD A. SHAFFER 著

张 铭 刘晓丹 译

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

PRENTICE HALL 出版公司

国外计算机科学教材系列

数据结构与算法分析

A Practical Introduction to DATA STRUCTURES AND ALGORITHM ANALYSIS

CLIFFORD A. SHAFFER 著

张 铭 刘晓丹 译

PRENTICE HALL 出版公司

電子工業出版社

内容提要

本书把数据结构原理和算法分析技术有机地结合在一起，系统地介绍了各种类型的数据结构和各种排序、检索方法。作者还引入了一些比较高级的数据结构与算法分析技术，并介绍了可计算性理论的一般知识。作者采用了能更自然地体现抽象数据类型概念的C++语言作为算法描述语言，并注意对每一种数据结构的不同存储方法与有关算法进行比较分析。

本书概念清楚，逻辑性强，内容新颖，可作为大专院校计算机科学与技术系相关专业学生的教材或参考书，也可供从事计算机工程的技术人员参考。

丛 书 名：国外计算机科学教材系列
原 书 名：A Practical Introduction to DATA STRUCTURES AND ALGORITHM ANALYSIS
书　　名：**数据结构与算法分析**
著　　者：CLIFFORD A. SHAFFER
译　　者：张　铭　刘晓丹
责任编辑：宋杏珍
印 刷 者：北京市天竺颖华印刷厂
出版发行：电子工业出版社出版、发行　URL：http://www.phei.com.cn
北京市海淀区万寿路 173 信箱　邮编　100036　发行部电话 68214070
经　　销：各地新华书店经销
开　　本：787×1092 1/16　印张：22　字数：511 千字
版　　次：1998 年 8 月第 1 版　1998 年 8 月第 1 次印刷
定　　价：35.00 元
印　　数：7000 册
书　　号：ISBN 7-5053-4595-8/TP·2176
著作权合同登记号　图字：01-97-1862
凡购买电子工业出版社的图书，如有缺页、倒页、脱页者，本社发行部负责调换

出版说明

计算机科学的迅速发展是20世纪科学发展史上最伟大的事件之一。从1946年第一台笨重而体积庞大的计算机的发明至今，仅仅半个多世纪，计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面，成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中，我国正在加强计算机科学的高等教育，着眼于为下一世纪培养高素质的计算机人才，以适应信息社会加速度发展的需要。当前，全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容，而作为与计算机科学密切相关的计算机、通信、信息等专业，更是在酝酿着教学的全面革新，以期规划出一整套面向21世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此，我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材，洋为中用，以更好地服务于国内的高校教育。

美国Prentice Hall出版公司是享誉世界的高校教材出版商，自1913年公司成立以来，即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用，其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者，如：道格拉斯·科默(Douglas Comer)，安德鲁·坦尼伯姆(Andrew Tanenbaum)，威廉·斯大林(William Stallings)……几十年来，他们的著作教育了一批批不同肤色的莘莘学子，使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量，电子工业出版社和Prentice Hall出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学，上海地区的上海交通大学、复旦大学，南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士，有积累了几十年教学经验的教授和博士生导师，还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材，在我们还是第一次。除缺乏经验之外，还由于我们对计算机科学的发展，对中国高校计算机教育特点认识的不足，致使在选题确定、翻译、出版等工作中，肯定存在许多遗憾和不足之处，恳请广大师生和其他读者提出批评、建议。

电子工业出版社
URL：http://www.phei.com.cn
Prentice Hall出版公司
URL：http://www.prenhall.com

译者序

数据结构以及算法分析是计算机专业十分重要的基础课，计算机科学各领域及各种应用软件都要使用相关的数据结构和算法。

这本书有三大特色：首先，它把数据结构原理和算法分析技术有机地结合在一起。每一种数据结构与其相关的算法都有其时间空间开销和效率，这就涉及到算法分析技术。当面临一个新的设计问题时，设计者要彻底掌握怎样权衡时间空间开销的方法和设计有效算法的技术，以适应问题的需要。把数据结构和算法分析揉合在一本教材中，有助于读者根据问题的性质选择合理的数据结构，并对时间空间复杂性进行必要的控制以设计高效的算法。

其次，本书对数据结构和算法的描述，采用了当前流行的面向对象的C++语言，这是程序员最广泛使用的语言。这样，程序员就可以把本书的许多算法直接应用于将来的实际项目中。尽管数据结构和算法设计本质上还是很底层的东西，并不像软件工程大型项目设计对面向对象方法具有直接的依赖性，也许有人认为并不需要采用面向对象的高级技术来描述底层的算法，但是采用C++语言能够更好地体现抽象数据类型的概念，从而更本质地描述数据结构和算法。为了使本书清晰易懂，作者有意回避了C++的某些重要特性，因此本书中的算法都是很容易读懂的。

第三，作者十分注意对每一种数据结构的存储结构及其有关算法的来龙去脉、不同方法进行利弊分析，这种"授人以渔"的策略使读者在今后设计和应用数据结构时能够全面地考虑各种因素，选择最佳方案。另外，作者还引入了一些比较高级的数据结构，可以开阔读者的视野。作者提供的参考书目也是颇有价值的。

本书内容分为四大部分。第一部分是准备知识，包括第1章到第3章。介绍了一些基本概念和术语，以及基本的数学知识。

第二部分包括第4章到第7章，介绍了最基本的数据结构，依次为线性表(包括栈和队列)、二叉树、树和图。每种数据结构都从其数学特性入手，先介绍其抽象数据类型，然后再讨论其不同的存储方法，并且研究不同存储方法的可能算法。值得赞赏的是作者结合算法分析来讨论各种存储方法和算法的利弊，摒弃那些不适宜的方法，这样调动了读者思维并可从中学到考虑问题的方法。

作为最常用的算法，排序和检索历来是数据结构讨论的重点问题。在第三部分的第8章到第11章中作了详尽的讨论。排序算法最能够体现算法分析的魅力，它的算法速度要求非常高，而检索则考虑怎样提高检索速度，这往往是与存储方法有关的。书中介绍了几种高效的数据结构，例如自组织线性表、散列表、B树和B+树等，都具有极好的检索性能。

第四部分从第12章到第15章，介绍了数据结构的应用与一些高级主题。对提高程序员的算法分析能力具有重要的作用。

最后是附录部分。附录介绍了对于理解本书例子必要的一些C++语法，从而使得不懂C++语法的C和Pascal程序员能够更好地理解本书。

本书的前言及第1章至第8章由张铭翻译，第9章至第15章由刘晓丹翻译。限于水平难免有不妥之处，欢迎批评指正。

前　言

学习数据结构可以使我们学会编写效率更高的程序。也许有人要问："在新计算机速度日新月异的今天，为什么还需要高效的程序？"回答是我们的雄心与能力是一起增长的。现代计算技术在计算能力和存储容量上的革命仅仅提供了计算更复杂问题的有效工具，而程序的高效性要求是永远不会过时的。

程序高效性的要求不会也不应该与合理的设计和清晰的编码相矛盾。高效程序的设计基于良好的信息组织和优秀算法，而不是基于"编程小技巧"。一个程序员如果没有掌握设计简明清晰程序的基本原理，就不可能编写有效的程序。反过来说，简洁的程序需要合理的数据组织和清晰的算法。大多数计算机科学系设置课程时都认识到要培养良好的程序设计技能，首先应该强调基本的软件工程原理。因此，一旦程序员学会了设计和实现简明清晰的程序之原理，下一步就应该学习有效的数据组织和算法，以提高程序的效率。

途径：本书描述了许多表示数据的技术。这些技术包括以下的原则：

1. 每一种数据结构和每一个算法都有其时间空间的开销和效益。当面临一个新的设计问题时，设计者要彻底地掌握怎样权衡时空开销和算法有效性的方法，以适应问题的需要。这就需要懂得算法分析的原理，而且还需要了解所使用的物理介质的特性（例如，数据存储在磁盘上与存储在主存上，就有不同的考虑）。

2. 与开销和效益有关的是时空的权衡。例如，人们通常用时间来换取空间，或者相反地用空间来换取时间。程序员所面对的时空权衡普遍地存在于软件设计和实现的各个阶段，因此这个概念必须牢记在心。

3. 程序员应该充分了解一些现成的方法，以免作不必要的重复开发工作。因此，学生需要了解经常使用的数据结构和相关的算法。

4. 数据结构服从于应用的需要。学生们必须把应用需要放在第一位，然后寻找一个与实际应用相匹配的数据结构。要做到这一点，需要应用上述三条原则。

组织：数据结构和算法设计的书籍往往囿于下面这两种情形之一：一种是教材，一种是百科全书。有的书籍试图融合这两种编排，但通常是二者都没有组织好。本书是作为教材来编写的。我相信了解选择或设计解决问题的高效数据结构的基本原理是十分重要的，这比死记硬背书本内容重要得多。因此，我在本书中涵盖了大多数但不是所有标准的数据结构。为了阐述一些重要的原理，也包括了某些并非广泛使用的数据结构。另外，还介绍了一些相对较新但即将得到广泛应用的数据结构。

本书可以作为本科生一个学期的教学内容，也可作为专业技术人员的自学教材。读者应该具有编程的经验，最好学过相当于两个学期的结构化程序设计语言 Pascal 或 C。在第 2 章中，复习了一些数学预备知识，早已熟悉数学归纳法和递归的读者会容易掌握一些。

尽管本书应该一个学期完成，但书中超过了一个学期的内容。这可以为教师提供一些选择的余地。二年级的学生具有很少的基本数据结构和算法分析的背景，可以对他们详细地讲解第 1～12 章的内容，再从第 13 章中选择一些专题来讲解，我就是这样来给二年级学

生讲课的。背景知识更丰富的学生，可以先读第 1 章，跳过第 2 章中除参考书目之外的内容，简要地浏览第 3 章和第 4 章(请着重阅读第 4.1.3 小节)，然后详细阅读其余的章节。另外，教师可以根据程序设计实习的需要，选择第 13 章以后的的某些专题内容。

第 13 章是针对做较大的程序设计练习而编写的。我建议所有选修数据结构的学生，都应该做一些高级的树结构或其他较复杂的动态数据结构的上机实习，比如第 12 章中的跳跃表(Skip List)或稀疏矩阵。所有这些数据结构都不比二叉检索树更难，而且学完第 5 章后的学生都有能力来实现它们。

关于 C++：本书中所有示例程序都是用 C++ 来写的，但是我并不想难倒那些对 C++ 不熟悉的读者。在保持 C++ 优点的同时，使示例程序尽量简明、清晰。C++ 在本书中只是作为阐释数据结构的工具，而且实际上我们只用了 C++ 的一个小子集。特别是我用到了 C++ 隐蔽实现细节的特性，例如类(class)、私有成员(private class member)、构造函数(constructor)、析构函数(destructor)。这些特性支持了一个关键的概念：体现于抽象数据类型(abstract data type)中的逻辑设计与体现于数据结构中的物理实现的分离。

我在本书中没有讲授 C++ 语言的意图。只是提供了一个附录来解释 C++ 的一些基本语义和概念，这对读懂示例程序是必要的。另外还提供了通过匿名 FTP 得到本书中 C++ 源代码的途径。

为了使本书尽可能清晰易懂，我完全被回避了 C++ 的某些重要特性。有意地排除或尽量少使用一些特性，而这些特性是有经验的 C++ 程序员通常所使用的，比如类的层次(class hierarchy)、继承(inheritance)和虚函数(virtual function)。也很少使用运算符和函数的重载(operator and function overloading)。C 的原始语义比 C++ 所提供的某些类似功能要好一些。

虽然上述 C++ 的特性在实际程序中是正当的设计基础，但是它们只能掩盖而不是加强本书中所阐述的原理。例如，类的继承在避免重复编码和降低程序错误率方面很重要；但是从教育学的标准观点来看，类的继承在若干类中分散了数据元素的描述，从而使得程序更难理解。因此，当类的继承对阐述的观点有明显的作用时才使用它(例如第 5.3.1 小节)。避免代码重复和减少错误是很重要的目标，请不要把本书中的示例程序直接拷贝到你自己的程序中，而只是把它们看作是对数据结构原理的阐释。

最痛苦的选择是：我要决定在例题代码中是否使用模板(template)。考虑到它们的语义对于不熟悉 C++ 语言的人来说掩盖了其代码的含义，最后决定不使用模板。但是，有C++ 经验的人可以很容易地使用模板来修改示例程序。附录中提供了一个这种修改的例子。

本书中的示例程序提供了有关数据结构原理的具体描述，而不是一系列具有商业质量的类的实现。这些例子中所作的参数检查，比起从事商业软件设计的程序员在编程中所作的要少得多。某些参数检查是以调用 C 库函数 assert 的形式包含进来的。assert 的输入是一个布尔表达式，一旦这个表达式的值为假(FALSE)，程序就立即终止。这种功能在实际程序中通常认为是不必要的，但是对于理解数据结构怎样运作是十分有用的。

在示例程序中，我严格区分了“C++ 实现”和“伪码”(pseudocode)。一个标明“C++ 实现”的示例程序在一个以上的编译器中真正被编译过。伪码的示例通常具有与 C++ 接近的语法，但是典型地包含一行以上的更高级的描述。当我发现简单的、尽管并不十分精确的描述具有更好的教学效果时就使用了伪码。

几乎每一章都是以“深入学习导读”(Further Reading)一节来结束的。它并不是那一章内容的综合参考索引,而是通过这些书籍或文章提供给读者更广泛的信息和乐趣。有些情况下我也提供了某个知名计算机科学家的重要背景文章。

习题和项目设计:只靠读书是不能学会灵活地使用数据结构的。一定要通过编制实际的程序,比较不同的数据结构技术来观察在一个给定的条件下哪一种结构更加有效。另外,学生也需要通过编程序来提高算法分析与设计能力。这里提供了300多个练习题和项目设计的程序实习题,希望读者能够很好地利用它们。

与作者的联系方法以及相关资料的获取:本书中难免有一些错误,有些方面还有待进一步研究。作者非常欢迎读者指正,并提出建设性的意见。作者在因特网(Internet)上的电子函件(E-mail)地址是 shaffer@cs.vt.edu。也可以写信给以下的地址:

Cliff Shaffer
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

与本书有关的一套基于 LaTeX 系统的幻灯片材料可以通过 FTP 的匿名帐号(anonymous),从 ftp.prenhall.com 的目录中获得。该目录为:

pub/esm/computer_science.s-041/shaffer/ds/supplements/transparencies

例题的 C++ 代码可以从相同 FTP 站点下面的目录获得:

pub/esm/computer_science.s-041/shaffer/ds/supplements/code

弗吉尼亚技术学院二年级数据结构课程的 WWW 主页(Home Page)之 URL 为:

http://ei.cs.vt.edu/~cs2604

该网址上同时可以看到一个针对数据结构的可视化和图形化调试工具,称为 SWAN 系统。

本书是由 LaTeX 的写作系统来编排的,LaTeX 是 TeX 系统的一个软件包。参考书目(bibliography)是用 BibTeX 编排的,词汇表用 makeindex 来编写。图形主要是用 Xfig 来作的,但图 3.1 和图 10.5 实验性地使用了 Mathematica。

鸣谢:本书得到了许多友人的帮助。我想特别感谢其中的几位,他们对本书的出版贡献最大。对于没有被提及的朋友在此表示歉意。系主任 Jack Carroll 对本书给予了重要的精神上的帮助。弗吉尼亚技术学院在 1994 年秋季的学术研究假期中使得整个出书的事情成为可能,我是从那时开始着手准备的。Mike Keenan、Lenny Heath 和 Jeff Shaffer 对本书最初版本的内容提供了最有价值的意见。尤其是 Lenny Heath 多年来对于算法设计和分析以及怎样把二者讲授给学生,与我进行了深入的探讨。Layne Watson 提供了有关 Mathematica 的帮助,Bo Begole 提供了一些技术上的帮助。Mark Abrams 和 Dennis Kafura 回答了一些有关 C++ 的问题。

对于许多评阅了本书手稿的朋友,本人欠情甚深。这些评阅者是:Douglas Campbell (Brigham Young University), Vijay Kumar Garg(德克萨斯大学, University of Texas - Austin), Jim Miller(肯萨斯大学, University of Kansas), Bruce Maxim(密执根大学, University of Michigan - Dearborn), Jeff Parker(哈佛大学, Agile Networks/Harvard), Dana Richards(乔

治·梅森大学，Goerge Mason University）和 Jack Tan（休斯顿大学，University of Houston）。若不是他们的热心帮助，本书会出现更多技术上的错误，内容也将更肤浅。

没有 Prentice Hall 出版社众多朋友的帮助，不可能有本书的出版，因为作者不可能自己印出书来。因此，我要感谢 Laure Steele 和 Alan Apt 这两位编辑。感谢本书的责任编辑 Kathleen Caren 在本书接近出版的最忙乱的日子里，保持各个方面运作良好。感谢 Bill Zobrist 和 Bruce Gregory 使我着手此事。感谢 Prentice Hall 的 Linda Behrens 和 Phyllis Bregman 在本书出版过程中的帮助。可能还有许多没有被提及的 Prentice Hall 出版社的朋友，默默地提供了帮助。

十分感激 Hanan Samet 传授给我有关数据结构的知识。从他那里我学到了许多原理，当然本书中可能的错误并不是他的责任。感谢我的妻子 Terry 对我的爱和支持。最后，也是最重要的是要感谢这些年来选修数据结构的学生，是他们使我知道了在数据结构课程中什么是重要的而什么应该忽略，许多深入的问题也是他们提供的。这本书献给他们。

Clifford A. Shaffer

1996 年于弗吉尼亚州，布莱克斯堡（Blacksburg，Virginia）

目　录

第一部分　预备知识

第 1 章　数据结构和算法

信息的表示是计算机科学的基础。大多数计算机程序的主要目标与其说是完成运算，不如说是存储和检索信息。从存储空间和运行时间的实现角度来看，这些程序必须组织信息，以支持高效的信息处理过程。因此，研究数据结构和算法以有效地支持程序的实现就成了计算机科学的核心问题。

1.1　数据结构的原则

1.1.1　学习数据结构的必要性

有人可能认为，随着计算机功能的日益强大，程序的运行效率变得越来越不那么重要了。然而，计算机功能越强大，人们就越想去尝试更复杂的问题。而更复杂的问题需要更大的计算量，这就使得对高效程序的需求更加明显，工作越复杂就越偏离人们的日常经验。今天，计算机科学家必须训练出彻底理解隐藏在高效程序设计后面的一般原理的能力，因为他们日常的生活经验并不具备这些能力。

一般说来，一种数据结构就是一类普通数据的表示及其相关操作。从数据表示的观点来看，即使是存储在计算机中的一个整数或者一个浮点数字也是一个简单的数据结构。更典型地，一个数据结构被认为是一组数据项的组织或者结构。存储在数组中的一个有序整数表就是这种结构的一个例子。

如果具有足够的空间来存储一组数据项，总会有可能在这个数据项集合中查找出指定的数据项、打印数据项或将这些数据项处理成任何期望得到的顺序，或者更改任何特定数据项的值。因此，就有可能对任何数据结构施加所有必要的运算。然而，选择不同的数据结构可能造成巨大的差异：同样一个程序，可能在几秒钟内运行完毕，也可能需要几天时间才能完成运行。

毋庸置疑，人们编写程序是为了解决问题，这个道理本来不用作者再次提出。然而，在选择数据结构解决特定问题时，头脑中有这个不言而喻的道理是具有重要意义的。只有通过预先分析问题来确定必须达到的性能目标，才有希望挑选出正确的数据结构。有相当多的程序设计人员却忽视了这一分析过程，而直接选用某一个他们习惯使用的、但是与问题不相称的数据结构，结果设计出一个低效程序。相反，当使用简单的设计能够达到性能目标时，选用复杂的数据表示来改进这个程序也是没有道理的。