



Data Structures and Problem Solving Using C++, 2nd

数据结构与问题求解 (C++版) (第2版)

(美) Mark Allen Weiss 著
张丽萍 译



清华大学出版社

国外经典教材·计算机科学与技术

数据结构与问题求解 (C++版)

(第2版)

Mark Allen Weiss 著

张丽萍 译

清华大学出版社
北 京

Simplified Chinese edition copyright © 2005 by **PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.**

Original English language title from Proprietor's edition of the Work.

Original English language title: **Data Structures and Problem Solving Using C++, 2nd by Mark Allen Weiss,**
Copyright © 2000

EISBN:020161250X

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as **Addison-Wesley**

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 **Addison-Wesley** 授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2004-6477 号

版权所有, 翻印必究。举报电话: **010-62782989 13501256678 13801310933**

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

数据结构与问题求解(C++版)(第2版)/(美)维斯(Weiss, M.A.)著; 张丽萍译. —北京: 清华大学出版社, 2005. 8

原书名为: **Data Structures and Problem Solving Using C++**

ISBN 7-302-11166-9

I. 数… II. ①维… ②张… III. ①数据结构 ②C语言-程序设计 IV. ①TP311.12 ②TP312

中国版本图书馆 CIP 数据核字(2005)第 059578 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

责任编辑: 常晓波

封面设计: 立日新

印刷者: 北京嘉实印刷有限公司

装订者: 三河市李旗庄少明装订厂

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 47.75 字数: 1186 千字

版 次: 2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

书 号: ISBN 7-302-11166-9/TP·7380

印 数: 1~3000

定 价: 86.00 元

出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整 and 改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教學需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

国外经典教材·计算机科学与技术

编审委员会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
杨宗源	华东师范大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟样旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

序 言

本书是为计算机科学专业的第 2 学期的课程而编写的，从典型的《数据结构》（CS-2，即计算机专业第 2 学期）开始直到高级的数据结构和算法分析。

CS-2 课程的内容经历过一段时间的演变。尽管多数人都同意这样的主题安排，但在具体的细节上还是有较大的分歧。获得一致认可的主题是软件开发的原则，最突出的是封装和信息隐藏的概念。理论上，所有的 CS-2 课程都倾向于包含运行时分析、递归、基本的排序方法和初等数据结构。许多大学还开设了高级课程，主题涉及到高级的数据结构、算法、运行时分析。本书中的材料设计同时考虑到这两种级别，因此读者不必再另外购买其他教科书。

尽管如此，争论最激烈的还是 CS-2 中编程语言的选择以及其他几项必要的基本选择，包括：

- 是否这么早就介绍面向对象的设计或基于对象的设计。
- 对数学水平的要求。
- 在实现数据结构及其使用之间达到恰当的平衡点，以及与所选语言相关的编程细节。

笔者写本书的目的是，从抽象思维和问题求解的角度来介绍数据结构和算法。笔者试图覆盖所有与数据结构、分析及其 C++ 实现有关的重要内容，同时对那些理论上似乎很有意义但实际上很少使用的数据结构，则是避而远之。几乎不可能有哪本书能像本书一样在一门课程里讲述所有不同的数据结构，包括数据结构的使用。因此，笔者设计了本教材，以便让教师能够灵活地选择主题。教师需要在实践和理论之间寻求平衡，然后选择最适合课程需要的主题。正如此序言后面所讨论的那样，笔者对课本进行了细致的组织，尽可能地降低了各章之间的依赖性。

统一的方法

笔者的基本假设是基于任何语言的软件开发工具都有一个庞大的库，许多数据结构就是这些库的一部分。笔者预感到数据结构教学的重心将从实现转向使用。在本书中，笔者采用了一套独特的方法，将数据结构分成规范和实现，并充分利用已有的数据结构库，即标准模板库（Standard Template Library, STL）。

在第二部分将会有一章（第 7 章）单独讲述适合大多数应用程序的 STL 子集。第二部分还讲述了基本的分析技术、递归和排序。第三部分介绍使用 STL 数据结构的应用程序。直到第四部分已经使用数据结构之后，才开始介绍 STL 的实现。因为 STL 是 C++ 的一部分（较早的编译器则使用本教材中的 STL 代码，请参阅稍后的“代码可用性”部分），学生可以使用现有的软件组件来设计大型项目。

尽管本书中大量使用了 STL，但本书并非针对 STL 的专著，也不是专门讲述 STL 实

现的入门读物。本书的重点在于数据结构和基本的问题求解技术。当然，数据结构设计中使用的技术大都适用于 STL 的实现，因此在第四部分有几章介绍了 STL 的实现。然而，教师可以选择第四部分中较简单的实现，而不必讨论 STL 协议。第 7 章介绍了 STL，这对理解第三部分的代码很有必要。笔者只是使用了一些基本的 STL。

许多教师更喜欢定义、实现，然后使用每个数据结构的传统方法。由于第三部分和第四部分中的材料之间并不存在依赖关系，因此可以利用本书轻松地教授传统方法。

预备技能

阅读本书的学生应该了解一门面向对象或面向过程的编程语言。必须了解编程语言的基本特征，包括基元数据类型、运算符、控制结构、函数（方法）、输入和输出（但并不需要了解数组和类）。

已初步接触过 C++ 或 Java 的学生可能会觉得前两章的某些内容很简单。但其他部分所讲的 C++ 技术细节则相对比较深奥，在入门课程中可能不会讲到这些知识。

学习了其他语言课程的学生应该从第 1 章开始学习，并且应该仔细研读。还应该查阅附录 A，因为附录 A 中讨论了某些专属于 C++ 的语言问题。如果喜欢同时参阅一本 C++ 参考书，可以参考第 1 章中给出的推荐书目。

离散数学方面的知识对学习本书很有帮助，但并非绝对必要。本书给出了几个数学证明，但对于更复杂的证明，则提示读者复习有关的数学知识。第 8 章以及第 19~24 章需要具备一定的数学技能。教师可以轻松地选择跳过数学证明，而只介绍证明结果。本书中所有的数学证明都被清晰地标出来了，并与本书正文部分分开。

第二版修订摘要

1. 重新编写了第一部分的大部分内容。第 1 章中，不再介绍基元数组（将对基元数组的讨论移到附录 D 中）；取而代之的是矢量数组，此外还介绍了 `push_back`。此次修订版中的指针内容的位置后移。第 2 章，对材料进行了大幅调整和简化。第 3 章增加了模板材料。第 4 章，重写了继承方面的内容，使行文更加简洁明了。第 4 章结尾增加了较深奥的 C++ 技术细节，对于高级应用而言这些技术知识非常重要。
2. 第一部分还新增了一章用于讨论设计模式。介绍了几种基于对象的模式，包括 `Functor`、`Wrapper` 和 `Iterator`，并讨论了使用继承技术的模式，包括 `Observer`。
3. 以 STL 为基础重写了第二部分的“标准模板库”一章。此次修订版的第 7 章介绍了通用接口（与第一版相同）和 STL 接口。
4. 第三部分的代码基于 STL。有几处代码比第 1 版中的代码更具面向对象的特征。重新编写了 Huffman 代码示例。
5. 第四部分，重写了通用数据结构，使之更简单易懂。此外，第四部分每章结尾还适当介绍了简单的 STL 实现。实现的组件包括 `vector`、`list`、`stack`、`queue`、`set`、`map`、`priority_queue`，以及各种函数对象和算法。

C++

使用 C++ 既有好处也有坏处。C++ 类允许分离接口和实现，同时隐藏实现的内部细节。

C++类完全支持抽象表示法。C++的优势在于其在行业内的广泛应用。学生会发现他们正在学习的材料的实践性很强，可以帮助他们找到工作，这使学生有足够的动力坚持学完本课程。C++的一个缺点是它不是适合教学的完美语言，尤其是在辅助课程上，因此需要特别注意避免形成不良的编程实践。C++的另一个缺点是，它并非是一个稳定的语言，有许多不同的编译器。

作者原本打算以独立于语言的方式编写本书，只集中讨论一般原理（如数据结构中的理论），只是顺便提供一些 C++代码，但实际上不可能这样做。C++代码很复杂，学生需要查看完整的代码示例来理解其微妙之处。正如前面所提到的，附录 A 中提供了 C++的简单回顾。本书的第一部分介绍了 C++中与数据结构有关的一些更高级的特性。

C++语言有几个特征需要重点考虑：模板、继承、异常、命名空间和 C++最近新增的一些特性，如标准库。关于这部分内容，笔者是这样处理的：

- **模板** 模板的使用很广泛。有些教师对模板的使用持保留态度，乃是因为模板会使代码变得很复杂，但笔者之所以介绍模板，则是因为模板是任何成熟的 C++程序中的基本概念。
- **继承** 笔者对继承的使用相对较少，因为继承比较复杂，并且数据结构并非继承的用武之地。此次修订版关于继承方面的介绍比第 1 版中的更少。然而，还是有一章专门讨论继承，“设计模式”中也有部分内容涉及到基于继承的模式。大多数教师都希望避免讲述继承，而希望讲述继承的教师将会在本书中找到足够丰富的材料。
- **异常** 异常的语义已经标准化，异常在许多编译器中都可以运行。然而，C++中的异常常常涉及到拙劣的代码，或是会带来很大的麻烦（例如，与模板一起使用），或者很可能需要讨论继承。因此，笔者在本书中很少使用异常。但还是简单地介绍了异常，在某些情况下，代码中会意外地抛出异常。然而，在第三部分的所有代码中，笔者一般不会去捕获异常（标准库中的大多数类都不会引发异常）。
- **命名空间** 命名空间，是 C++中最近新增的特性，在许多编译器中都不能正常工作。笔者并不会使用命名空间，而是在必要的时候导入完整的 `std` 命名空间（尽管有点笨拙，但却可以在大多数编译器中正常工作）。附录 A 讨论了命名空间。
- **最近新增的语言特性** 书中通篇都使用了 `bool` 数据类型。使用了新的 `static_cast` 运算符，而不是传统的 `cast` 运算符。最后，笔者在适当的时候使用了 `explicit`。笔者几乎没有使用其他新增特性（例如，笔者总是试图避免使用 `typename`）。
- **标准库** 如前所述，全书通篇都使用了 STL，并在线提供了一个安全的版本（执行额外的边界检查），并在第四部分给出了具体的实现。我们还使用了 `string` 类和更新的 `istream` 类，后者是标准库中的一部分。

教程的组织结构

在本教材中，笔者在第 1.1 节介绍了 C++和面向对象编程（尤其是抽象），讨论了数组、指针和其他 C++主题，然后继续讨论类、模板和继承的语法和使用。这些章节中的材料都经过全新改写。此次修订版还新增了一章讨论设计模式。

第二部分讨论了 Big-Oh 和算法范例，包括递归和随机。有一章专门讲述排序，一章专门讨论基本的数据结构。笔者使用 STL 来介绍接口和数据结构的运行时。关于这一点，教师可能有几种方法来讲解其他材料，包括下面两种：

1. 第四部分中，介绍每种数据结构时都会讨论相应的实现（STL 版本或者更简单的版本）。教师可以要求学生以各种方式扩充类，如习题中所建议的那样。
2. 说明如何使用 STL 类，并在课程后面讲述实现。第三部分中的案例研究可用来支持该方法。每种现代的 C++ 编译器上都有完整的实现，旧编译器在 Internet 上也能提供完整的实现，教师可以在编程项目中使用 STL。书中还简单介绍了有关此方法的细节。

第五部分讲述了高级的数据结构，如分裂树、配对堆、不相交集数据结构，如果时间允许，也可以在后续课程中讲述这方面的知识。

每章的组织结构

第一部分包括 5 章，讲述本书中要用到的部分 C++ 高级特性。第 1 章介绍数组、字符串、指针、引用和结构；第 2 章介绍 C++ 中的类机制，开始讨论面向对象的编程；第 3 章讲解模板，继续讨论面向对象编程；第 4 章介绍继承的使用。在这几章中作者编写了几个组件，包括字符串和矢量组件。第 5 章讨论了一些基本的设计模式，重点放在基于对象的模式，如 Function 对象、Wrapper、Adapter、Iterator 和 Pair，其中有些模式（尤其是 Wrapper 模式）在本书后面将会用到。

第二部分的重点在于基本的算法和构建代码块。第 6 章中全面讨论了时间复杂度，提供了 Big-Oh 表示法，还讨论和分析了二分查找。第 7 章很关键，因为此章讲述了 STL，并直观地说明了每种数据结构所支持操作的运行时间。第四部分提供了这些数据结构的实现，包括 STL 方式和简化版本。STL 版本适用于最近的编译器。第 8 章首先通过推导介绍了数学证明的表示法，讲述了递归。还讨论了分治法、动态编程和回溯法。有一节介绍了在实现 RSA 加密系统中要用到的几种递归数学算法。对于许多学生来说，第 8 章的第二部分可能更适合继续学习。第 9 章描述、编写并分析了几种基本的排序算法，包括插入排序、Shell 排序、合并排序、快速排序，以及间接排序法。此外还提供了排序法的复杂度下限，并讨论了与排序算法选择有关的问题。最后，篇幅不长的第 10 章讨论了随机数，包括随机数的产生及其在随机算法中的使用。

第三部分提供了几个案例，每章都围绕一个公共主题进行组织。第 11 章通过研究几个游戏介绍了几项重要的技术。第 12 章通过研究一种检查对称符号的算法和经典的运算符优先级分析算法，讨论了堆栈的使用。第 12 章还提供了上述两种算法的完整实现代码。第 13 章讨论了用于文件压缩和交叉引用的基本实用程序，提供了两种实用程序的完整实现。第 14 章通过给出一个可视作模拟程序的问题全面讨论了模拟程序，并讨论了更经典的事件驱动模拟程序。最后，第 15 章阐述了如何使用数据结构来实现几种针对图形的最短路径算法。

第四部分介绍了数据结构的实现。提供了几种简单操作（插入、查找和删除元素）的实现。在某些示例中，还介绍了 C++ 语法更复杂的 STL 实现。在这一部分用到了一些数学知识，尤其是第 19~21 章，教师可根据需要跳过此部分内容。第 16 章提供了堆栈和队列

的实现。分别使用扩展数组和链表实现了这些数据结构。该章结尾讨论了 STL 版本的实现。第 17 章介绍了常见的链表, 首先简单地介绍了单向链表, 然后在第 17 章结尾介绍了更复杂的、使用双向链表的 STL 版本。第 18 章介绍了树, 并阐述了基本的遍历方案。第 19 章详细讨论了二叉查找树的几种实现, 首先, 给出基本的二叉查找树, 然后派生出支持顺序统计的二叉查找树。第 19 章还讨论了 AVL 树, 但没有给出具体的实现, 而是实现了更具实践价值的红黑树和 AA 树, 然后实现了 STL set 和 STL map, 最后, 探讨了 B 树。第 20 章讨论了哈希表, 然后在给出较简单的方法之后给出了二次探查法的实现。第 21 章介绍了二叉堆, 讨论了堆排序和外部排序, 实现了 STL priority_queue。

第五部分包含适合高级课程和一般参考的材料。里面的算法甚至适合一年级学生, 然而, 为全面起见, 还包含了较深的数学分析, 这些内容可能超出一年级学生的范畴。第 22 章介绍了伸展树, 伸展树是一种二叉查找树, 在实际应用中的性能可能非常优秀, 在需要优先队列的应用程序中则可与二叉堆媲美。第 23 章介绍了支持合并操作的优先队列, 提供了配对堆的实现。最后, 第 24 章探讨了经典的不相交集合数据结构。

附录包含附加的 C++ 参考材料。附录 A 介绍了较复杂的 C++ 主题, 包括一些不常见的运算符、I/O 和最近的语言变化。附录 B 列出了运算符及其优先级。附录 C 总结了一些 C++ 库。附录 D 为想了解 Vector 类和 String 类内部细节的读者介绍了基元数组和字符串。

各章之间的依赖关系

一般而言, 大多数章节与其他章节没有依赖关系。然而, 下面各章之间存在明显的依赖关系。

- **第一部分** 首先应该将前 3 章作为一个整体来学习。在第 4 章, 笔者简单地讨论了继承技术。有些教师希望全面地讲述继承, 但了解基本的继承知识可能已经够用, 还可以避免与继承有关的、复杂的 C++ 问题。关于基于对象的模式 (例如包装程序和函数对象) 可以在第 5 章讨论模板之后讨论, 也可以根据需要在后面进行讨论。第四部分有关排序算法的内容中用到过其中的一些模式。
- **第 6 章 (算法分析)** 本章应该在第 7 章和第 9 章之前讲解。递归 (第 8 章) 则可以在第 6 章之前讲解, 但教师必须跳过某些内容以避免低效的递归。
- **第 7 章 (标准模板库)** 此章可以提前讲解, 也可以与第三部分或第四部分中的材料结合起来讲解。
- **第 8 章 (递归)** 第 8.1~8.3 节中的材料应该在讨论递归排序算法、树、tic-tac-toe 案例和最短路径算法之前讲解。否则, 也可以考虑讲解 RSA 加密系统、动态编程和回溯法之类的材料 (除非已讨论 tic-tac-toe)。
- **第 9 章 (排序算法)** 此章应该在第 6 章和第 8 章之后讲解。然而, 可以跳过第 6 章和第 8 章讲解 Shell 排序。Shell 排序不是递归的 (因此没有必要学习第 8 章), 对 Shell 排序的运行时间的严格分析非常复杂, 本书不予讨论 (因此几乎没有必要学习第 6 章)。
- **第 16 章 (堆栈和队列) 和第 17 章 (列表)** 这两章之间的顺序可以随意。然而, 笔者更倾向于先讲解第 16 章, 因为笔者相信第 16 章讲解了较简单的链表示例。
- **第 18 章 (树) 和第 19 章 (二叉查找树)** 这两章之间的顺序可以随意, 也可以

同时讲解。

独立的章节

其他章节之间没有或者几乎没有任何依赖关系。

- **第 10 章（随机）** 关于随机数的材料可以根据需要随时讲解。
- **每三部分（应用程序）** 第 11~15 章可以结合 STL（第 7 章）讲解，也可以在之后讲解，这 5 章之间几乎可以不分顺序。这几章中偶尔会引用前面的一些章节。这些引用包括：第 11.2 节（tic-tac-toe）引用了第 8.7 节的相关论述，第 13.2 节（交叉引用的生成）引用了类似于第 12.1 节（对称符号检查）中的词法分析。
- **第 20 章（散列表）和第 21 章（优先级队列）** 这两章可以安排在任何地方进行讲解。
- **第五部分（高级数据结构）** 第 22~24 章中的材料是自包含的，通常在后续课程中讲解。

数学

笔者曾试图提供在第 2 学期课程中使用的严密的数学证明，强化理论并为后续的、需要进一步分析的课程做准备。然而，这些内容都以定理的形式与正文区分开，在某些情况下，甚至单独构成一节（或一小节）。这样，教师在不强调理论的课程中就可以跳过这些内容。

在任何情况下，定理的证明对于理解定理的含义都不是必需的。这也是又一个接口（定理陈述）与实现（证明）分离的例子。有些数学材料乃是沿袭传统，例如第 8.4 节（递归的数学应用）就可以跳过不讲，这样不会影响学生对余下几章的理解。

课程组织

本课程教学的一个关键问题是决定如何组织使用第二部分到第四部分中的材料。第一部分中的材料比较深奥，学生应该编写一两个程序来亲身实践类和广义类的设计、实现和测试，也许还可以通过使用继承机制来体会面向对象的设计。第 6 章讨论了 Big-Oh 表示法。有一个习题要求学生编写一个短小的程序并通过分析比较运行时间。

按照分离的原则，第 7 章的主要思想是：不同的数据结构支持不同的访问方案，具有不同的效率。任何案例（除了 tic-tac-toe 示例以外）都可以用来示范数据结构的应用。这样，学生可以看到数据结构并知道如何使用数据结构，而不是如何有效地实现数据结构。这确实是一种分离法。这种观察事物的方式能够极大地提高学生的抽象思维能力。学生也可以提供 STL 组件的简单实现（第 7 章的习题中提供了一些建议），并了解到现有 STL 中高效数据结构实现和低效数据结构实现之间的区别。也可以要求学生扩展案例，但再次强调，不必要求学生了解数据结构的全部细节。

数据结构的有效实现可以放在后面介绍，递归的介绍则可以根据教师的感觉而定，但务必在二叉树之前讲解。排序法可安排在递归之后的任意时间。在此，可以使用相同的案例继续本课程的教学，并亲身修改数据结构的实现。举例来说，学生可以体验各种形式的均衡二叉树。

喜欢传统方法的教师只需在介绍第四部分中的数据结构实现之后介绍第三部分中的一个案例。再重复一次，本书在章节设计上尽可能保持各章之间的独立性。

练习

习题的风格多样，笔者提供了 4 种风格的习题：基本的“简答题”提出一个简单的问题，或者要求动手编写本书中的某个算法。“理论题”提的问题可能需要数学分析或要求提出具有理论意义的解决方案。“实践题”包含简单的编程问题，包括有关语法的问题或特殊的代码问题。最后，“编程项目”则是对课堂作业的延伸。

教学特征

- 旁注用于突出显示重要的主题。
- “学习目标”列出了重要的术语及其定义，以及页面引用。
- 每章结尾的“常见错误”部分列出了经常出现的错误。
- 大多数章节都给出了进一步学习的参考材料。

代码可用性

本书中的代码绝对可以运行，并且已在众多平台上测试过。可从笔者的主页下载：<http://www.fiu.edu/~weiss>。请务必查阅 Readme 文件以了解有关编译器依赖性和 bug 修复的信息。每章结尾的“Internet 资源”部分列出了每章代码的文件名。

教师资源指南

本书附有教师指南，提供了几种可用的材料，包括测试题、作业和提纲样例。此外还提供了选择题的答案。教师可与当地的 Addison Wesley Longman 经销商联系或发送电子邮件至 aw.cse@awl.com 来了解本书的销售信息。此指南仅供教师参考，不予销售。

致 谢

帮助我准备此书的人有许多许多。其中有很多,笔者曾在第1版和相关书目 *Data Structures and Problem Solving Using Java* 中专门致谢过。此外,还有很多人,曾经发送过电子邮件,指出书中的错误或解释的不一致问题(这一版我已尽力修正了此问题),但由于人数真的太多,笔者无法在此一一列出。

对于这本书,我要感谢 Addison Wesley Longman 的所有人:编辑 Susan Hartman 和助理编辑 Katherine Harutunian,曾帮助我在 C++材料的组织上做出艰难的决定,本书得以面市,他们有很大的功劳。责任校对 Jerrold Moore 和校对人员,提出了许多有益的建议,帮助改进本书。Diana Coe 奉献了一个精美的封面设计。一如既往,Michael Hirsch 在市场方面做出了卓越的奉献。特别感谢制作编辑 Pat Mahtani 和 Shepherd 公司的 Lynn Steines,感谢他们在协调整个项目方面做出的杰出贡献。

还要感谢各地的评论家,他们提出了许多宝贵的意见,有许多已经融入到本书当中。他们是:

Zhengxin Chen, 奥马哈 Nebraska 大学
Arlan DeKock, Missouri-Rolla 大学
Andrew Duchowski, Clemson 大学
Seth Copen Goldstein, Carnegie Mellon 大学
G. E. Hedrick, 俄克拉荷马州立大学
Murali Medidi, Northern Arizona 大学
Chris Nevison, Colgate 大学
Gurpur Prabhu, 爱荷华州立大学
Donna Reese, 密西西比州立大学
Gurdip Singh, 堪萨斯州立大学
Michael Stinson, Central Michigan 大学
Paul Wolfgang, Temple 大学

本书中的部分材料摘自笔者的教材 *Efficient C Programming: A Practical Approach* (Prentice-Hall, 1995), 并获得出版商的许可,并在适当的地方加上了章尾参考。

我的主页 <http://www.cs.fiu.edu/~weiss> 包括更新后的源代码、勘误表和发送错误报告的链接。

M.A.W.
Miami, Florida
1999 年 9 月

目 录

第一部分 对象和 C++

第 1 章 数组、指针和结构	1	参考文献	28
1.1 什么是指针、数组和结构	1	第 2 章 对象和类	30
1.2 数组和字符串	2	2.1 什么是面向对象编程	30
1.2.1 头等对象与次等对象的对比	2	2.2 类的基本语法	31
1.2.2 使用 Vector	3	2.2.1 类成员	31
1.2.3 调整 Vector 大小	5	2.2.2 附加的构造函数语法和访问函数	33
1.2.4 push_back 大小与容量	7	2.2.3 接口和实现的分离	35
1.2.5 参数传递机制	7	2.2.4 析构函数、复制构造函数和赋值运算符(=)	38
1.2.6 常量基元数组	9	2.2.5 默认的构造函数	43
1.2.7 多维数组	9	2.3 附加的 C++ 类特性	43
1.2.8 标准库类型 string	9	2.3.1 调整后的构造函数中的初始化与赋值	47
1.3 C++ 中的指针语法	10	2.3.2 类型转换	48
1.4 动态内存管理	14	2.3.3 运算符重载	49
1.4.1 new 运算符	15	2.3.4 输入、输出和友元	52
1.4.2 垃圾收集与 delete	15	2.4 一些常用术语	54
1.4.3 过期指针、双重删除及其他	16	2.4.1 避免使用友元	54
1.5 引用变量	17	2.4.2 静态类成员	55
1.6 结构	19	2.4.3 整型类常量的陷阱	55
1.6.1 指向结构的指针	21	2.5 异常	56
1.6.2 外部数据与内部数据、深复制与浅复制	21	2.6 String 类	57
1.6.3 非邻接链表: 链表	23	2.7 要点重述: 进行了哪些调用? 哪些采用了默认行为	65
小结	24	2.8 组合	66
学习目标	24	小结	67
常见错误	25	学习目标	68
网上资源	26	常见错误	69
练习	26	Internet 资源	70
简答题	26	练习	70
实践题	28		
编程项目	28		

简答题	70	4.2.1 可视性规则	98
理论题	71	4.2.2 构造函数和基类初始化	98
编程项目	72	4.2.3 添加成员	99
参考文献	75	4.2.4 覆盖方法	101
第 3 章 模板	76	4.2.5 静态绑定和动态绑定	101
3.1 模板的概念	76	4.2.6 默认的构造函数、复制构造 函数、复制赋值运 算符和析构函数	103
3.2 函数模板	76	4.2.7 构造函数和析构函数 virtual 或非 virtual	104
3.3 排序函数模板	78	4.2.8 抽象方法和抽象类	105
3.4 类模板	81	4.3 例子: 扩展 Shape 类	108
3.4.1 MemoryCell 模板	81	4.4 微妙的 C++ 细节	112
3.4.2 实现 vector 类模板	85	4.4.1 参数的静态绑定	113
3.5 模板的模板: matrix 类	87	4.4.2 默认参数	114
3.5.1 数据成员、构造函数和 基本附件	88	4.4.3 派生类方法隐藏 基类方法	114
3.5.2 operator []	89	4.4.4 覆盖方法的兼容返回 类型	115
3.5.3 析构函数、复制赋值和 复制构造函数	89	4.4.5 私有继承	116
3.6 Fancy 模板	89	4.4.6 友元	116
3.6.1 多平台参数	89	4.4.7 值调用与多态并不混淆	117
3.6.2 默认的模板参数	90	4.5 多重继承	117
3.6.3 保留字 typename	90	小结	118
3.7 与模板有关的 bug	90	学习目标	119
3.7.1 错误消息和改变的规则	91	常见错误	119
3.7.2 模板匹配算法	91	Internet 资源	120
3.7.3 模板中的嵌套类	91	练习	120
3.7.4 类模板中的静态成员	91	简答题	120
小结	91	实践题	122
学习目标	91	编程项目	122
常见错误	92	参考文献	122
Internet 资源	92	第 5 章 设计模式	123
练习	93	5.1 模式的概念	123
简答题	93	5.2 Functor (函数对象)	124
实践题	93	5.3 适配器和包装器	129
编程项目	93	5.3.1 指针包装器	129
第 4 章 继承	94	5.3.2 常数引用包装器	134
4.1 什么是继承	94		
4.2 继承的基本知识	97		

5.3.3 适配器更改接口	135	学习目标	148
5.4 迭代器	136	常见错误	149
5.4.1 迭代器设计 1	137	Internet 资源	149
5.4.2 迭代器设计 2	139	练习	150
5.4.3 基于继承的迭代器和 factory	139	简答题	150
5.5 合成(对)	144	理论题	150
5.6 观察者	144	实践题	150
小结	148	编程项目	152
		参考文献	152

第二部分 算法和构建代码块

第 6 章 算法分析	153	7.2 堆栈和队列	183
6.1 什么是算法分析	153	7.2.1 堆栈	184
6.2 算法运行时间的例子	156	7.2.2 堆栈和计算机语言	185
6.3 最大连续子数列和问题	157	7.2.3 队列	186
6.3.1 直观的 $O(N^3)$ 算法	158	7.3 容器和迭代器	187
6.3.2 改进的 $O(N^2)$ 算法	160	7.3.1 容器	187
6.3.3 线性算法	161	7.3.2 迭代器	188
6.4 一般的 Big-Oh 规则	164	7.4 STL 算法	189
6.5 对数	167	7.4.1 STL 函数对象	189
6.6 静态查找问题	169	7.4.2 二分查找法	191
6.6.1 顺序查找	169	7.4.3 排序	193
6.6.2 折半查找	169	7.5 实现带有迭代器的 vector	193
6.6.3 插值查找	172	7.6 顺序表和链表	195
6.7 算法分析的检验	173	7.6.1 list 类	195
6.8 Big-Oh 分析的限制	174	7.6.2 堆栈和队列	196
小结	174	7.7 集合	197
学习目标	174	7.8 映射	199
常见错误	175	7.9 优先队列	200
Internet 资源	175	小结	203
练习	176	学习目标	204
简答题	176	常见错误	204
理论题	177	Internet 资源	205
实践题	179	练习	205
编程项目	179	简答题	205
参考文献	180	理论题	205
		实践题	206
		编程项目	206
		参考文献	208
第 7 章 标准模板库	182		
7.1 简介	182		

第 8 章 递归	209	9.5 归并排序	257
8.1 递归的概念	209	9.5.1 排过序的数组的线性 时间合并	257
8.2 背景知识: 数学归纳法	210	9.5.2 归并排序算法	259
8.3 基本递归	212	9.6 快速排序	261
8.3.1 以任意基数打印数字	213	9.6.1 快速排序算法	261
8.3.2 递归算法有效的原因	215	9.6.2 快速排序的分析	263
8.3.3 递归算法的作用原理	216	9.6.3 支点的选择	265
8.3.4 递归不宜太多	217	9.6.4 分组策略	267
8.3.5 树	218	9.6.5 同支点相等的键	268
8.3.6 附加例子	219	9.6.6 中值划分	269
8.4 数值应用	223	9.6.7 小数组	270
8.4.1 模运算	223	9.6.8 C++快速排序例程	270
8.4.2 模幂运算	224	9.7 排序选择	272
8.4.3 最大公约数和乘法 逆元素	225	9.8 排序的下限	274
8.4.4 RSA 密码系统	228	9.9 间接排序	275
8.5 分治算法	230	9.9.1 使用指针将 Comparable 副本数减少为 $2N$	275
8.5.1 最大邻近子序列和问题	230	9.9.2 避免附加数组	276
8.5.2 基本分治递归分析	233	小结	278
8.5.3 分治法运行时间的 一般上限	235	学习目标	279
8.6 动态规划	237	常见错误	279
8.7 回溯法	241	Internet 资源	279
小结	244	练习	280
学习目标	244	简答题	280
常见错误	245	理论题	280
Internet 资源	245	实践题	281
练习	246	编程项目	282
简答题	246	参考文献	283
理论题	246		
实践题	247		
编程项目	248		
参考文献	249		
第 9 章 排序算法	250		
9.1 排序为何重要	250	第 10 章 随机	285
9.2 预备知识	251	10.1 为什么我们需要随机数	285
9.3 插入排序和其他简单排序的分析	252	10.2 随机数生成器	286
9.4 希尔排序	254	10.3 非均匀随机数	290
		10.4 生成随机排列	291
		10.5 随机算法	293
		10.6 随机素数测试	295
		小结	298
		学习目标	298