<<算法（第4版）>>

图书基本信息

书名：<<算法（第4版）>>

13位ISBN编号：9787115293800

10位ISBN编号：7115293805

出版时间：2012-10

出版时间：人民邮电出版社

作者：[美] Robert Sedgewick,[美]Kevin Wayne

页数：636

字数：1115000

译者：谢路云

前言

本书力图研究当今最重要的计算机算法并将一些最基础的技能传授给广大求知者。
它适合用做计算机科学进阶教材，面向已经熟悉了计算机系统并掌握了基本编程技能的学生。
本书也可用于自学，或是作为开发人员的参考手册，因为书中实现了许多实用算法并详尽分析了它们的性能特点和用途。
这本书取材广泛，很适合作为该领域的入门教材。
算法和数据结构的学习是所有计算机科学教学计划的基础，但它并不只是对程序员和计算机系的学生有用。
任何计算机使用者都希望计算机能运行得更快一些或是能解决更大规模的问题。
本书中的算法代表了近50年来的大量优秀研究成果，是人们工作中必备的知识。
从物理中的N体模拟问题到分子生物学中的基因序列问题，我们描述的基本方法对科学研究而言已经必不可少；从建筑建模系统到模拟飞行器，这些算法已经成为工程领域极其重要的工具；从数据库系统到互联网搜索引擎，算法已成为现代软件系统中不可或缺的一部分。
这仅是几个例子而已，随着计算机应用领域的不断扩张，这些基础方法的影响也会不断扩大。
在开始学习这些基础算法之前，我们先要熟悉全书中都将会用到的栈、队列等低级抽象的数据类型。
然后依次研究排序、搜索、图和字符串方面的基础算法。
最后一章将会从宏观角度总结全书的内容。
独特之处本书致力于研究有实用价值的算法。
书中讲解了多种算法和数据结构，并提供了大量相关的信息，读者应该能有信心在各种计算环境下实现、调试并应用它们。
本书的特点涉及以下几个方面。
算法　书中均有算法的完整实现，并讨论了程序在多个样例上的运行状况。
书中的代码都是可以运行的程序而非伪代码，因此非常便于投入使用。
书中程序是用Java语言编写的，但其编程风格方便读者使用其他现代编程语言重用其中的大部分代码来实现相同算法。
数据类型我们在数据抽象上采用了现代编程风格，将数据结构和算法封装在了一起。
应用每一章都会给出所述算法起到关键作用的应用场景。
这些场景多种多样，包括物理模拟与分子生物学、计算机与系统工程学，以及我们熟悉的数据压缩和网络搜索等。
学术性我们非常重视使用数学模型来描述算法的性能。
我们用模型预测算法的性能，然后在真实的环境中运行程序来验证预测。
广度本书讨论了基本的抽象数据类型、排序算法、搜索算法、图及字符串处理。
我们在算法的讨论中研究数据结构、算法设计范式、归纳法和解题模型。
这将涵盖20世纪60年代以来的经典方法以及近年来产生的新方法。
我们的主要目标是将今天最重要的实用算法介绍给尽可能广泛的群体。
这些算法一般都十分巧妙奇特，20行左右的代码就足以表达。
它们展现出的问题解决能力令人叹为观止。
没有它们，创造计算智能、解决科学问题、开发商业软件都是不可能的。
本书网站本书的一个亮点是它的配套网站algs4.cs.princeton.edu。
这一网站面向教师、学生和专业人士，免费提供关于算法和数据结构的丰富资料。
一份在线大纲　包含了本书内容的结构并提供了链接，浏览起来十分方便。
全部实现代码　书中所有的代码均可以在这里找到，且其形式适合用于程序开发。
此外，还包括算法的其他实现，例如高级的实现、书中提及的改进的实现、部分习题的答案以及多个应用场景的客户端代码。
我们的重点是用真实的应用环境来测试算法。
习题与答案　网站还提供了一些附加的选择题（只需要一次单击便可获取答案）、很多算法应用的例

子、编程练习和答案以及一些有挑战性的难题。
动态可视化　书是死的，但网站是活的，在这里我们充分利用图形类演示了算法的应用效果。
课程资料　网站包含和本书及网上内容对应的一整套幻灯片，以及一系列编程作业、核对表、测试数据和备课手册。
相关资料链接　网站包含大量的链接，提供算法应用的更多背景知识以及学习算法的其他资源。
我们希望这个站点和本书互为补充。
一般来说，建议读者在第一次学习某种算法或是希望获得整体概念时看书，并把网站作为编程时的参考或是在线查找更多信息的起点。
作为教材本书为计算机科学专业进阶的教材，涵盖了这门学科的核心内容，并能让学生充分锻炼编程、定量推理和解决问题等方面的能力。
一般来说，此前学过一门计算机方面的先导课程就足矣，只要熟悉一门现代编程语言并熟知现代计算机系统，就都能够阅读本书。
虽然本书使用Java实现算法和数据结构，但其代码风格使得熟悉其他现代编程语言的人也能看懂。
我们充分利用了Java的抽象性（包括泛型），但不会依赖这门语言的独门特性。
书中涉及的多数数学知识都有完整的讲解（少数会有延伸阅读），因此阅读本书并不需要准备太多数学知识，不过有一定的数学基础当然更好。
应用场景都来自其他学科的基础内容，同样也在书中有完整介绍。
本书涉及的内容是任何准备主修计算机科学、电气工程、运筹学等专业的学生应了解的基础知识，并且对所有对科学、数学或工程学感兴趣的学生也十分有价值。
背景介绍这本书意在接续我们的一本基础教材《Java程序设计：一种跨学科的方法》，那本书对计算机领域做了概括性介绍。
这两本书合起来可用做两到三个学期的计算机科学入门课程教材，为所有学生在自然科学、工程学和社会科学中解决计算问题提供必备的基础知识。
本书大部分内容来自Sedgewick的算法系列图书。
本质上，本书和该系列的第1版和第2版最接近，但还包含了作者多年教学和学习的经验。
Sedgewick的《C算法（第3版）》、《C++算法（第3版）》、《Java算法（第3版）》更适合用做参考书或是高级课程的教材，而本书则是专门为大学一、二年级学生设计的一学期教材，也是最新的基础入门书或从业者的参考书。
致谢本书的编写花了近40年时间，因此想要一一列出所有参与人是不可能的。
本书的前几版一共列出了好几十人，其中包括（按字母顺序）Andrew Appel、Trina Avery、Marc Brown、Lyn Dupré、PhilippeFlajolet、Tom Freeman、Dave Hanson、Janet Incerpi、Mike Schidlowsky、Steve Summit和Chris VanWyk。
我要感谢他们所有人，尽管其中有些人的贡献要追溯到几十年前。
至于第4版，我们要感谢试用了本书样稿的普林斯顿及其他院校的数百名学生，以及通过本书网站发表意见和指出错误的世界各地的读者。
我们还要感谢普林斯顿大学对于高质量教学的坚定支持，这是本书得以面世的基础。
Peter Gordon几乎从本书写作之初就提出了很多有用的建议，这一版奉行的“归本溯源”的指导思想也是他最早提出的。
关于第4版，我们要感谢Barbara Wood认真又专业的编辑工作，Julie Nahil对生产过程的管理，以及Pearson出版公司中为本书的付梓和营销辛勤工作的朋友。
所有人都在积极地追赶进度，而本书的质量并没有受到丝毫影响。

<<算法（第4版）>>

内容概要

《算法（第4版）》全面讲述算法和数据结构的必备知识，具有以下几大特色。

1、 算法领域的经典参考书：Sedgewick畅销著作的最新版，反映了经过几十年演化而成的算法核心知识体系

2、内容全面：全面论述排序、搜索、图处理和字符串处理的算法和数据结构，涵盖每位程序员应知应会的50种算法

3、全新修订的代码：全新的Java实现代码，采用模块化的编程风格，所有代码均可供读者使用

4、与实际应用相结合：在重要的科学、工程和商业应用环境下探讨算法，给出了算法的实际代码，而非同类著作常用的伪代码

5、富于智力趣味性：简明扼要的内容，用丰富的视觉元素展示的示例，精心设计的代码，详尽的历史和科学背景知识，各种难度的练习，这一切都将使读者手不释卷

6、科学的方法：用合适的数学模型精确地讨论算法性能，这些模型是在真实环境中得到验证的

7、与网络相结合：配套网站algs4.cs.princeton.edu提供了本书内容的摘要及相关的代码、测试数据、编程练习、教学课件等资源

作者简介

Robert Sedgewick，斯坦福大学博士，导师为Donald E. Knuth，从1985年开始一直担任普林斯顿大学计算机科学系教授，曾任该系主任，也是Adobe Systems公司董事会成员，曾在Xerox PARC、国防分析研究所（Institute for Defense Analyses）和法国国家信息与自动化研究所（INRIA）从事研究工作。
他的研究方向包括解析组合学、数据结构和算法的分析与设计、程序可视化等。

Kevin Wayne，康奈尔大学博士，普林斯顿大学计算机科学系高级讲师，研究方向包括算法的设计、分析和实现，特别是图和离散优化。

<<算法（第4版）>>

书籍目录

章节摘录

版权页： 插图： 警告：这段类型转换的用例代码和1.3.2.2节所示的有所不同。
你可能会以为需要使用Object而非Stack。
在使用泛型时，Java会在编译时检查类型的安全性，但会在运行时抛弃所有这些信息。
因此在运行时语句右侧就变成了Stack（）或者只剩下了Stack（），因此我们必须将它们转化为Stack（）。
问 在栈为空时调用pop（）会发生什么？
答 这取决于实现。
对于我们在算法1.2中给出的实现，你会得到一个NullPointerException异常。
对于我们在本书的网站上给出的实现，我们会抛出一个运行时异常以帮助用户定位错误。
一般来说，在应用广泛的代码中这类检查越多越好。
问 既然有了链表，为什么还要学习如何调整数组的大小？
答 我们还将会学习若干抽象数据类型的示例实现，它们需要使用数组来实现一些链表难以实现的操作。
ResizingArrayStack是控制它们的内存使用的样板。
问 为什么将Node声明为嵌套类？
为什么使用private？
答 将Node声明为私有的嵌套类之后，我们可以将Node的方法和实例变量的访问范围限制在包含它的类中。
私有嵌套类的一个特点是只有包含它的类能够直接访问它的实例变量，因此无需将它的实例变量声明为public或是private。
专业背景较强的读者注意：非静态的嵌套类也被称为内部类，因此从技术上来说我们的Node类也是内部类，尽管非泛型的类也可以是静态的。
问 当我输入javac Stack.java运行算法1.2和其他程序时，我发现了Stack.class和Stack$Node.class两个文件。第二个文件是做什么用的？
答 第二个文件是为内部类Node创建的。
Java的命名规则会使用$分隔外部类和内部类。
问 Java标准库中有栈和队列吗？
答 有，也没有。
Java有一个内置的库，叫做java.util.Stack，但你需要栈的时候请不要使用它。
它新增了几个一般不属于栈的方法，例如获取第i个元素。
它还允许从栈底添加元素（而非栈顶），所以它可以被当做队列使用！
尽管拥有这些额外的操作看起来可能很有用，但它们其实是累赘。
我们使用某种数据类型不仅仅是为了获得我们能够想象的各种操作，也是为了准确地指定我们所需要的操作。
这么做的主要好处在于系统能够防止我们执行一些意外的操作。
java.util.Stack的API是宽接口的一个典型例子，我们通常会极力避免出现这种情况。

编辑推荐

Sedgewick之巨著，与高德纳TAOCP一脉相承几十年多次修订，经久不衰的畅销书涵盖所有程序员必须掌握的50种算法

<<算法（第4版）>>

版权说明