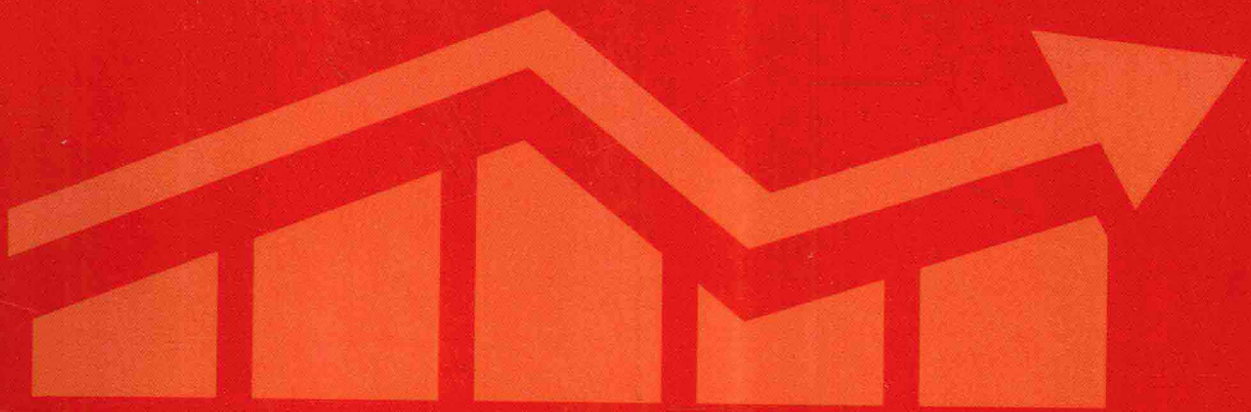
TURING 图灵程序设计丛书

PEARSON

- 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

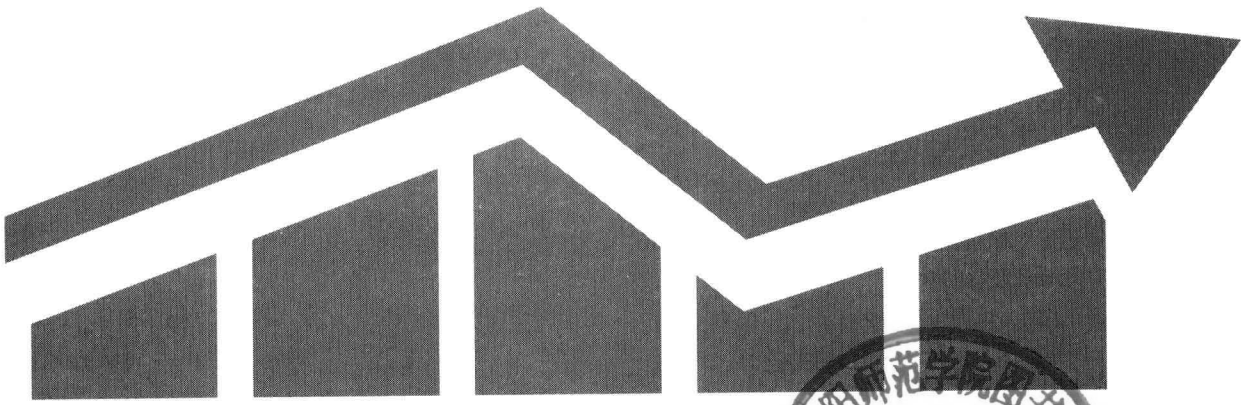


算法 Algorithms Fourth Edition （第4版）

[美] Robert Sedgewick
Kevin Wayne 著
谢路云 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书



算法 Algorithms (第4版) Fourth Edition

[美] Robert Sedgewick
Kevin Wayne 著

谢路云 译

人民邮电出版社
北京

图书在版编目（CIP）数据

算法 : 第4版 / (美) 塞奇威克 (Sedgewick,R.) , (美) 韦恩 (Wayne,K.) 著 ; 谢路云译. -- 北京 : 人民邮电出版社, 2012.10
(图灵程序设计丛书)
书名原文: Algorithms, Fourth Edition
ISBN 978-7-115-29380-0

Ⅰ. ①算… Ⅱ. ①塞… ②韦… ③谢… Ⅲ. ①电子计算机－算法理论 Ⅳ. ①TP301.6

中国版本图书馆CIP数据核字(2012)第220659号

内 容 提 要

本书作为算法领域经典的参考书，全面介绍了关于算法和数据结构的必备知识，并特别针对排序、搜索、图处理和字符串处理进行了论述。第4版具体给出了每位程序员应知应会的50个算法，提供了实际代码，而且这些Java代码实现采用了模块化的编程风格，读者可以方便地加以改造。配套网站提供了本书内容的摘要及更多的代码实现、测试数据、练习、教学课件等资源。

本书适合用做大学教材或从业者的参考书。

图灵程序设计丛书
算法（第4版）

◆ 著 [美] Robert Sedgewick Kevin Wayne
译 谢路云
责任编辑 朱 巍
执行编辑 丁晓昀 刘美英

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092 1/16
印张：40.5 彩插 12
字数：1115千字 2012年10月第1版
印数：1－5 000册 2012年10月北京第1次印刷
著作权合同登记号 图字：01-2011-5236号
ISBN 978-7-115-29380-0

定价：99.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223
反盗版热线：(010)67171154

版权声明

谨以此书献给Adam、Andrew、Brett、Robbie，并特别感谢Linda。

——Robert Sedgewick

献给Jackie和Alex。

——Kevin Wayne

译 者 序

在计算机领域，算法是一个永恒的主题。即使仅把算法入门方面的书都摆出来，国内外的加起来怕是能铺满整个天安门广场。在这些书中，有几本尤其与众不同，本书就是其中之一。

本书是学生的良师。在翻译的过程中我曾无数次感叹："要是当年我能拥有这本书那该多好！"应该说本书是为在校学生量身打造的。没有数学基础？没关系，只要你在高中学过了数学归纳法，那么书中95%以上的数学内容你都可以看得懂，更何况书中还辅以大量图例。没学过编程？没关系，第1章会给大家介绍足够多的Java知识，即使你不是计算机专业的学生，也不会遇到困难。整本书的内容编排循序渐进，由易到难，前后呼应，足见作者的良苦用心。没有比本书更专业的算法教科书了。

本书是老师的好帮手。如果老师们还只能照本宣科，只能停留在算法本身一二三四的阶段，那就已经大大落后于这个时代了。算法并不仅仅是计算的方法，探究算法的过程反映出的是我们对这个世界的认知方法：是唯唯诺诺地将课本当做圣经，还是通过"实验—失败—再实验"循环的锤炼？数学是保证，数据是验证。本书通过各种算法，从各个角度，多次说明了这个道理，这也正是第1章是全书内容最多的一章的原因。希望每一位读者都不要错过第1章。无论你有没有编程基础，都会从中得到有益的启示。

本书是程序员的益友。在工作了多年之后，快速排序、霍夫曼编码、KMP等曾经熟悉的概念在你脑中是不是已经凋零成了一个个没有内涵的名词？是时候重新拾起它们了。无论是为手头的工作寻找线索，还是为下一份工作努力准备，这些算法基础知识都是你不能跳过的。本书强调软件工程中的最佳实践，特别适合已有工作经验的程序员朋友。所有的算法都是先有API，再有实现，之后是证明，最后是数据。这种先接口后实现、强调测试的做法，无疑是在工作中摸爬滚打多年的程序员最熟悉的。

本书也有一些遗憾，比如没有介绍动态规划这样重要的思想。但是瑕不掩玉，它仍然是最好的入门级算法书。我强烈地希望能够把本书翻译成中文，但同时也诚惶诚恐，如履薄冰，担心自己的水平不足以准确传达原文的意思。翻译的过程虽然辛苦，但我觉得非常值得。感谢人民邮电出版社图灵公司给了我这个机会，感谢编辑和审稿专家的细心检查。同时感谢我的妻子朱天的全力支持。译者水平有限，bug在所难免，还请读者批评指正。

谢路云

2012.9.17

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。

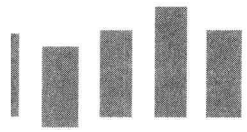
致谢

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、Philippe Flajolet、Tom Freeman、Dave Hanson、Janet Incerpi、Mike Schidlowsky、Steve Summit和Chris Van Wyk。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。

目　录



第1章 基 础

本书的目的是研究多种重要而实用的*算法*，即适合用计算机实现的解决问题的方法。和算法关系最紧密的是*数据结构*，即便于算法操作的组织数据的方法。本章介绍的就是学习算法和数据结构所需要的基本工具。

首先要介绍的是我们的*基础编程模型*。本书中的程序只用到了 Java 语言的一小部分，以及我们自己编写的用于封装输入输出以及统计的一些库。1.1 节总结了相关的语法、语言特性和书中将会用到的库。

接下来我们的重点是*数据抽象*并定义*抽象数据类型*（ADT）以进行模块化编程。在 1.2 节中我们介绍了用 Java 实现抽象数据类型的过程，包括定义它的*应用程序编程接口*（API）然后通过 Java 的类机制来实现它以供各种用例使用。

之后，作为重要而实用的例子，我们将学习三种基础的抽象数据类型：*背包*、*队列*和*栈*。1.3 节用数组、变长数组和链表实现了背包、队列和栈的 API，它们是全书算法实现的起点和样板。

性能是算法研究的一个核心问题。1.4 节描述了分析算法性能的方法。我们的基本做法是*科学式*的，即先对性能提出假设，建立数学模型，然后用多种实验验证它们，必要时重复这个过程。

我们用一个*连通性*问题作为例子结束本章，它的解法所用到的算法和数据结构可以实现经典的 union-find 抽象数据结构。

算法

编写一段计算机程序一般都是实现一种已有的*方法*来解决某个问题。这种方法大多和使用的编程语言无关——它适用于各种计算机以及编程语言。是这种方法而非计算机程序本身描述了解决问题的步骤。在计算机科学领域，我们用*算法*这个词来描述一种有限、确定、有效的并适合用计算机程序来实现的解决问题的方法。算法是计算机科学的基础，是这个领域研究的核心。

要定义一个算法，我们可以用自然语言描述解决某个问题的过程或是编写一段程序来实现这个过程。如发明于 2300 多年前的*欧几里德算法*所示，其目的是找到两个数的最大公约数：

自然语言描述

计算两个非负整数 p 和 q 的最大公约数：若 q 是 0，则最大公约数为 p。否则，将 p 除以 q 得到余数 r，p 和 q 的最大公约数即为 q 和 r 的最大公约数。

Java 语言描述

```
Public static int gcd(int p, int q)
{
    if (q == 0) return p;
    int r = p % q;
    return gcd(q, r);
}
```

欧几里德算法

如果你不熟悉欧几里德算法，那么你应该在学习了 1.1 节之后完成练习 1.1.24 和练习 1.1.25。在本书中，我们将用计算机程序来描述算法。这样做的重要原因之一是可以更容易地验证它们是否如所要求的那样有限、确定和有效。但你还应该意识到用某种特定语言写出一段程序只是表达一个算法的一种方法。数十年来本书中许多算法都曾被表达为多种编程语言的程序，这正说明每种算法都是适合于在任何计算机上用任何编程语言实现的方法。

我们关注的大多数算法都需要适当地组织数据，而为了组织数据就产生了*数据结构*，数据结构也是计算机科学研究的核心对象，它和算法的关系非常密切。在本书中，我们的观点是数据结构是算法的副产品或是结果，因此要理解算法必须学习数据结构。简单的算法也会产生复杂的数据结构，相应地，复杂的算法也许只需要简单的数据结构。本书中我们将会研讨许多数据结构的性质，也许
4 本书就应该叫《*算法与数据结构*》。

当用计算机解决一个问题时，一般都存在多种不同的方法。对于小型问题，只要管用，方法的不同并没有什么关系。但是对于大型问题（或者是需要解决大量小型问题的应用），我们就需要设计能够有效利用时间和空间的方法了。

学习算法的主要原因是它们能节约非常多的资源，甚至能够让我们完成一些本不可能完成的任务。在某些需要处理上百万个对象的应用程序中，设计优良的算法甚至可以将程序运行的速度提高数百万倍。在本书中我们将在多个场景中看到这样的例子。与此相反，花费金钱和时间去购置新的硬件可能只能将速度提高十倍或是百倍。无论在任何应用领域，精心设计的算法都是解决大型问题最有效的方法。

在编写庞大或者复杂的程序时，理解和定义问题、控制问题的复杂度和将其分解为更容易解决的子问题需要大量的工作。很多时候，分解后的子问题所需的算法实现起来都比较简单。但是在大多数情况下，某些算法的选择是非常关键的，因为大多数系统资源都会消耗在它们身上。本书的焦点就是这类算法。我们所研究的基础算法在许多应用领域都是解决困难问题的有效方法。

计算机程序的共享已经变得越来越广泛，尽管书中涉及了许多算法，我们也只实现了其中的一小部分。例如，Java 库包含了许多重要算法的实现。但是，实现这些基础算法的简化版本有助于我们更好地理解、使用和优化它们在库中的高级版本。更重要的是，我们经常需要重新实现这些基础算法，因为在全新的环境中（无论是硬件的还是软件的），原有的实现无法将新环境的优势完全发挥出来。在本书中，我们的重点是用最简洁的方式实现优秀的算法。我们会仔细地实现算法的关键部分，并尽最大努力揭示如何进行有效的底层优化工作。

5 为一项任务选择最合适的算法是困难的，这可能会需要复杂的数学分析。计算机科学中研究这种问题的分支叫做*算法分析*。通过分析，我们将要学习的许多算法都有着优秀的理论性能；而另一些我们则只是根据经验知道它们是可用的。我们的主要目标是学习典型问题的各种有效算法，但也会注意比较不同算法之间的性能差异。不应该使用资源消耗情况未知的算法，因此我们会时刻关注算法的期望性能。

本书框架

接下来概述一下全书的主要内容，给出涉及的主题以及本书大致的组织结构。这组主题触及了尽可能多的基础算法，其中的某些领域是计算机科学的核心内容，通过对这些领域的深入研究，我们找出了应用广泛的基本算法，而另一些算法则来自计算机科学和相关领域比较前沿的研究成果。总之，本书讨论的算法都是数十年来研发的重要成果，它们将继续在快速发展的计算机应用中扮演重要角色。

第 1 章　基础

它讲解了在随后的章节中用来实现、分析和比较算法的基本原则和方法，包括 Java 编程模型、数据抽象、基本数据结构、集合类的抽象数据类型、算法性能分析的方法和一个案例分析。

第 2 章　排序

有序地重新排列数组中的元素是非常重要的基础算法。我们会深入研究各种排序算法，包括插入排序、选择排序、希尔排序、快速排序、归并排序和堆排序。同时我们还会讨论另外一些算法，它们用于解决几个与排序相关的问题，例如优先队列、选举以及归并。其中许多算法会成为后续章节中其他算法的基础。

第 3 章　查找

从庞大的数据集中找到指定的条目也是非常重要的。我们将会讨论基本的和高级的查找算法，包括二叉查找树、平衡查找树和散列表。我们会梳理这些方法之间的关系并比较它们的性能。

第 4 章　图

图的主要内容是对象和它们的连接，连接可能有权重和方向。利用图可以为大量重要而困难的问题建模，因此图算法的设计也是本书的一个主要研究领域。我们会研究深度优先搜索、广度优先搜索、连通性问题以及若干其他算法和应用，包括 Kruskal 和 Prim 的最小生成树算法、Dijkstra 和 Bellman-Ford 的最短路径算法。

6

第 5 章　字符串

字符串是现代应用程序中的重要数据类型。我们将会研究一系列处理字符串的算法，首先是对字符串键的排序和查找的快速算法，然后是子字符串查找、正则表达式模式匹配和数据压缩算法。此外，在分析一些本身就十分重要的基础问题之后，这一章对相关领域的前沿话题也作了介绍。

第 6 章　背景

这一章将讨论与本书内容有关的若干其他前沿研究领域，包括科学计算、运筹学和计算理论。我们会介绍性地讲一下基于事件的模拟、B 树、后缀数组、最大流量问题以及其他高级主题，以帮助读者理解算法在许多有趣的前沿研究领域中所起到的巨大作用。最后，我们会讲一讲搜索问题、问题转化和 NP 完全性等算法研究的支柱理论，以及它们和本书内容的联系。

学习算法是非常有趣和令人激动的，因为这是一个历久弥新的领域（我们学习的绝大多数算法都还不到“五十岁”，有些还是最近才发明的，但也有一些算法已经有数百年的历史）。这个领域不断有新的发现，但研究透彻的算法仍然是少数。本书中既有精巧、复杂和高难度的算法，也有优雅、朴素和简单的算法。在科学和商业应用中，我们的目标是理解前者并熟悉后者，这样才能掌握这些有用的工具并学会*算法式思考*，以迎接未来计算任务的挑战。

7

1.1 基础编程模型

我们学习算法的方法是用 Java 编程语言编写的程序来实现算法。这样做是出于以下原因：

- 程序是对算法精确、优雅和完全的描述；
- 可以通过运行程序来学习算法的各种性质；
- 可以在应用程序中直接使用这些算法。

相比用自然语言描述算法，这些是重要而巨大的优势。

这样做的一个缺点是我们要使用特定的编程语言，这会使分离算法的思想和实现细节变得困难。我们在实现算法时考虑到了这一点，只使用了大多数现代编程语言都具有且能够充分描述算法所必需的语法。

我们仅使用了 Java 的一个子集。尽管我们没有明确地说明这个子集的范围，但你也会看到我们只使用了很少的 Java 特性，而且会优先使用大多数现代编程语言所共有的语法。我们的代码是完整的，因此希望你能下载这些代码并用我们的测试数据或是你自己的来运行它们。

我们把描述和实现算法所用到的语言特性、软件库和操作系统特性总称为*基础编程模型*。本节以及 1.2 节会详细说明这个模型，相关内容自成一体，主要是作为文档供读者查阅，以便理解本书的代码。我们的另一本入门级的书籍 *An Introduction to Programming in Java: An Interdisciplinary Approach* 也使用了这个模型。

作为参考，图 1.1.1 所示的是一个完整的 Java 程序。它说明了我们的基础编程模型的许多基本特点。在讨论语言特性时我们会用这段代码作为例子，但可以先不用考虑代码的实际意义（它实现了经典的*二分查找*算法，并在*白名单过滤*应用中对算法进行了检验，请见 1.1.10 节）。我们假设你具备某种主流语言编程的经验，因此你应该知道这段代码中的大多数要点。图中的注释应该能够解答你的任何疑问。因为图中的代码某种程度上反映了本书代码的风格，而且对各种 Java 编程惯例和语言构造，在用法上我们都力求一致，所以即使是经验丰富的 Java 程序员也应该看一看。

8

1.1.1 Java 程序的基本结构

一段 Java 程序（*类*）或者是一个*静态方法*（*函数*）*库*，或者*定义了一个数据类型*。要创建静态方法库和定义数据类型，会用到下面五种语法，它们是Java语言的基础，也是大多数现代语言所共有的。

- *原始数据类型*：它们在计算机程序中精确地定义*整数*、*浮点数*和*布尔值*等。它们的定义包括取值范围和能够对相应的值进行的*操作*，它们能够被组合为类似于数学公式定义的*表达式*。
- *语句*：语句通过创建*变量*并对其赋值、控制运行流程或者引发副作用来进行计算。我们会使用六种语句：*声明*、*赋值*、*条件*、*循环*、*调用*和*返回*。
- *数组*：数组是多个同种数据类型的值的集合。
- *静态方法*：静态方法可以封装并重用代码，使我们可以用独立的模块开发程序。
- *字符串*：字符串是一连串的字符，Java 内置了对它们的一些操作。
- *标准输入 / 输出*：标准输入输出是程序与外界联系的桥梁。
- *数据抽象*：数据抽象封装和重用代码，使我们可以定义非原始数据类型，进而支持面向对象编程。

我们将在本节学习前五种语法，数据抽象是下一节的主题。

运行 Java 程序需要和操作系统或开发环境打交道。为了清晰和简洁，我们把这种输入命令执行程序的环境称为*虚拟终端*。请登录本书的网站去了解如何使用虚拟终端，或是现代系统中许多其他高级的编程开发环境的使用方法。

在例子中，BinarySearch 类有两个静态方法 `rank()` 和 `main()`。第一个方法 `rank()` 含有四条语句：两条声明语句，一条循环语句（该语句中又有一条赋值语句和两条条件语句）和一条返回语句。

第二个方法 main() 包含三条语句：一条声明语句、一条调用语句和一个循环语句（该语句中又包含一条赋值语句和一条条件语句）。

要执行一个 Java 程序，首先需要用 javac 命令编译它，然后再用 java 命令运行它。例如，要运行 BinarySearch，首先要输入 javac BinarySearch.java（这将生成一个叫 BinarySearch.class 的文件，其中含有这个程序的 Java 字节码）；然后再输入 java BinarySearch（接着是一个白名单文件名）把控制权移交给这段字节码程序。为了理解这段程序，我们接下来要详细介绍原始数据类型和表达式，各种 Java 语句、数组、静态方法、字符串和输入输出。

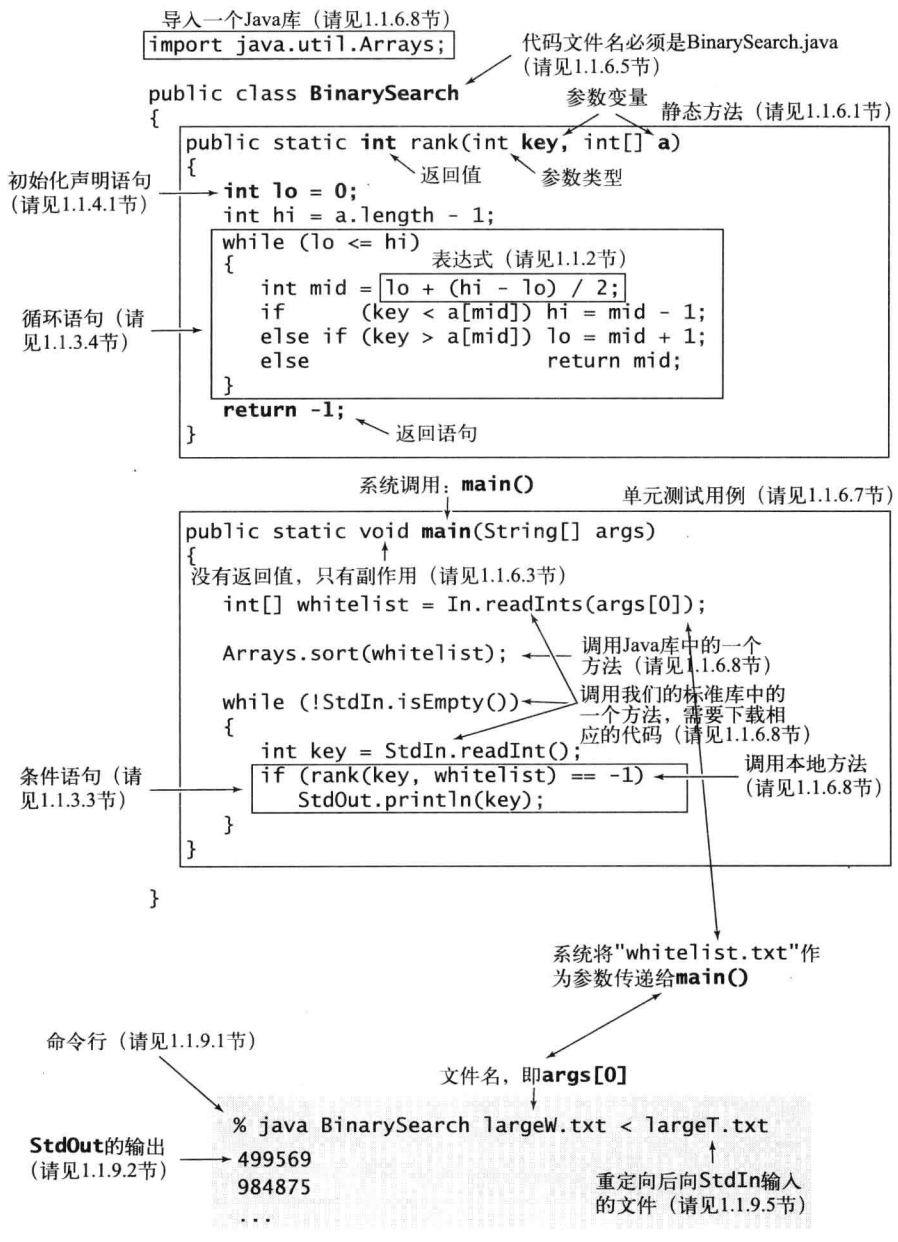


图 1.1.1 Java 程序及其命令行的调用