

# 编程珠玑（第2版·修订版）

作者：[美]乔恩·本特利（Jon Bentley） 著

目录

[封面](#)

[扉页](#)

[版权](#)

[版权声明](#)

[译者简介](#)

[译者序](#)

[前言](#)

[第一部分 基础](#)

[第1章 开篇](#)

[1.1 一次友好的对话](#)

[1.2 准确的问题描述](#)

[1.3 程序设计](#)

[1.4 实现概要](#)

[1.5 原理](#)

[1.6 习题](#)

[1.7 深入阅读](#)

[第2章 啊哈！算法](#)

[2.1 三个问题](#)

[2.2 无处不在的二分搜索](#)

[2.3 基本操作的威力](#)

[2.4 排序](#)

[2.5 原理](#)

[2.6 习题](#)

[2.7 深入阅读](#)

[2.8 变位词程序的实现（边栏）](#)

[第3章 数据决定程序结构](#)

[3.1 一个调查程序](#)

[3.2 格式信函编程](#)

[3.3 一组示例](#)

[3.4 结构化数据](#)

[3.5 用于特殊数据的强大工具](#)

[3.6 原理](#)

[3.7 习题](#)

[3.8 深入阅读](#)

[第4章 编写正确的程序](#)

[4.1 二分搜索的挑战](#)

[4.2 编写程序](#)

[4.3 理解程序](#)

[4.4 原理](#)

[4.5 程序验证的角色](#)

[4.6 习题](#)

[4.7 深入阅读](#)

[第5章 编程小事](#)

[5.1 从伪代码到C程序](#)

[5.2 测试工具](#)

[5.3 断言的艺术](#)

[5.4 自动测试](#)

[5.5 计时](#)

[5.6 完整的程序](#)

[5.7 原理](#)

[5.8 习题](#)

[5.9 深入阅读](#)

[5.10 调试（边栏）](#)

[第二部分 性能](#)

[第6章 程序性能分析](#)

[6.1 实例研究](#)

[6.2 设计层面](#)

[6.3 原理](#)

[6.4 习题](#)

[6.5 深入阅读](#)

[第7章 粗略估算](#)

[7.1 基本技巧](#)

[7.2 性能估计](#)

[7.3 安全系数](#)

[7.4 Little定律](#)

[7.5 原理](#)

[7.6 习题](#)

[7.7 深入阅读](#)

[7.8 日常生活中的速算（边栏）](#)

[第8章 算法设计技术](#)

[8.1 问题及简单算法](#)

[8.2 两个平方算法](#)

[8.3 分治算法](#)

[8.4 扫描算法](#)

[8.5 实际运行时间](#)

[8.6 原理](#)  
[8.7 习题](#)  
[8.8 深入阅读](#)  
[第9章 代码调优](#)  
[9.1 典型的故事](#)  
[9.2 急救方案集锦](#)  
[9.3 大手术——二分搜索](#)  
[9.4 原理](#)  
[9.5 习题](#)  
[9.6 深入阅读](#)  
[第10章 节省空间](#)  
[10.1 关键在于简单](#)  
[10.2 示例问题](#)  
[10.3 数据空间技术](#)  
[10.4 代码空间技术](#)  
[10.5 原理](#)  
[10.6 习题](#)  
[10.7 深入阅读](#)  
[10.8 巨大的节省（边栏）](#)  
[第三部分 应用](#)  
[第11章 排序](#)  
[11.1 插入排序](#)  
[11.2 一种简单的快速排序](#)  
[11.3 更好的几种快速排序](#)  
[11.4 原理](#)  
[11.5 习题](#)  
[11.6 深入阅读](#)  
[第12章 取样问题](#)  
[12.1 问题](#)  
[12.2 一种解决方案](#)  
[12.3 设计空间](#)  
[12.4 原理](#)  
[12.5 习题](#)  
[12.6 深入阅读](#)  
[第13章 搜索](#)  
[13.1 接口](#)  
[13.2 线性结构](#)  
[13.3 二分搜索树](#)  
[13.4 用于整数的结构](#)  
[13.5 原理](#)  
[13.6 习题](#)  
[13.7 深入阅读](#)  
[13.8 一个实际搜索问题（边栏）](#)  
[第14章 堆](#)  
[14.1 数据结构](#)  
[14.2 两个关键函数](#)  
[14.3 优先级队列](#)  
[14.4 一种排序算法](#)  
[14.5 原理](#)  
[14.6 习题](#)  
[14.7 深入阅读](#)  
[第15章 字符串](#)  
[15.1 单词](#)  
[15.2 短语](#)  
[15.3 生成文本](#)  
[15.4 原理](#)  
[15.5 习题](#)  
[15.6 深入阅读](#)  
[第1版跋](#)  
[第2版跋](#)  
[附录A 算法分类](#)  
[附录B 估算测试](#)  
[附录C 时空开销模型](#)  
[附录D 代码调优法则](#)  
[附录E 用于搜索的C++类](#)  
[部分习题提示](#)  
[部分习题答案](#)  
[索引](#)

PEARSON

编程珠玑 第2版 修订版

[美]Jon Bentley 著

黄倩 钱丽艳 译

刘田 审校

Programming Pearls Second Edition

人民邮电出版社

北京

图书在版编目 (CIP) 数据

编程珠玑/(美)本特利 (Bentley,J.)著;黄倩,钱丽艳译.--2版(修订本).--北京:人民邮电出版社,2015.1

书名原文: Programming pearls,second edition

ISBN 978-7-115-35761-8

I.①编... II.①本...②黄...③钱... III.①程序设计 IV.①TP311.1

中国版本图书馆CIP数据核字(2014)第266600号

内容提要

本书是计算机科学方面的经典名著。书的内容围绕程序设计人员面对的一系列实际问题展开。作者Jon Bentley以其独有的洞察力和创造力,引导读者理解这些问题并学会解决方法,而这些正是程序员实际编程生涯中至关重要的。本书的特色是通过一些精心设计的有趣而又颇具指导意义的程序,对实用程序设计技巧及基本设计原则进行了透彻而睿智的描述,为复杂的编程问题提供了清晰而完备的解决思路。本书对各个层次的程序员都具有很高的阅读价值。

◆著 [美]Jon Bentley

译 黄倩 钱丽艳

审校 刘田

责任编辑 杨海玲

责任印制 张佳莹 彭志环

◆人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆开本: 720×960 1/16

印张: 17.5

字数: 320千字 2015年1月第2版

印数: 1-6000册 2015年1月北京第1次印刷

著作权合同登记号 图字: 01-2006-7038号

定价: 39.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

## 版权声明

Authorized translation from the English language edition,entitled Programming Pearls,Second Edition,0201657880 by Jon Bentley,published by Pearson Education,Inc.,publishing as Addison Wesley,Copyright © 2000 by Lucent Technologies.

All rights reserved.No part of this book may be reproduced or transmitted in any form or by any means,electronic or mechanical,including photocopying,recording or by any information storage retrieval system,without permission from Pearson Education,Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD.and POSTS & TELECOM PRESS Copyright © 2015.

本书中文简体字版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

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。

版权所有，侵权必究。

## 译者简介

黄倩 中国科学院计算技术研究所博士研究生，毕业于南京大学，目前主要从事视频处理等方面的研究工作。

钱丽艳 北京大学信息科学技术学院基础实验教学研究所软件实验室主任、工程师，毕业于国防科技大学，目前主要从事数值计算、程序设计等方面的研究工作。

### 审稿人简介

刘田 北京大学信息科学技术学院软件研究所副教授、中国电子学会电路与系统分会图论与系统优化专业委员会秘书长、中国计算机学会和中国电子学会高级会员，毕业于中国科学技术大学，目前主要从事算法分析和计算复杂度、量子信息处理等方面的研究工作，翻译出版了多部国外著名离散数学和计算理论教材。

## 译者序

本书作者Jon Bentley是美国著名的程序员和计算机科学家，他于20世纪70年代前后在很有影响力的《ACM通讯》（Communications of the ACM）上以专栏的形式连续发表了一系列短文，成功地总结和提炼了自己在长期的计算机程序设计实践中积累下来的宝贵经验。这些短文充满了真知灼见，而且文笔生动、可读性强，对于提高职业程序员的专业技能很有帮助，因此该专栏大受读者欢迎，成为当时该学术期刊的王牌栏目之一。可以想象当时的情形颇似早年金庸先生在《明报》上连载其武侠小说的盛况。后来在ACM的鼓励下，作者经过仔细修订和补充整理，对各篇文章的先后次序做了精心编排，分别在1986年和1988年结集出版了Programming Pearls（《编程珠玑》）和More Programming Pearls（《编程珠玑（续）》）这两本书，二者均成为该领域的名著。《编程珠玑（第2版）》在2000年问世，书中的例子都改用C语言书写，并多处提到如何用C++和Java中的类来实现。《编程珠玑（续）》虽未再版，例子多以Awk语言写成，但其语法与C相近，容易看懂。

作者博览群书，旁征博引，无论是计算机专业的专业名著，如《计算机程序设计艺术》，还是普通的科普名著，如《啊哈！灵机一动》，都在作者笔下信手拈来、娓娓道出，更不用说随处可见的作者自己的真知灼见了。如果说《计算机程序设计艺术》这样的巨著代表了程序员们使用的“坦克和大炮”一类的重型武器，这两本书则在某种程度上类似于鲁迅先生所说的“匕首与投枪”一类的轻型武器，更能满足职业程序员的日常需要。或者说前者是武侠小说中提高内力修为的根本秘籍，后者是点拨临阵招数的速成宝典，二者同样都是克敌制胜的法宝，缺一不可。在无止境地追求精湛技艺这一点上，程序员、数学家和武侠们其实是相通的。

在美国，这两本书不仅被用作大学低年级数据结构与算法课程的教材，还用作高年级算法课程的辅助教材。例如，美国著名大学麻省理工学院的电气工程与计算机科学开放式核心课程算法导论就将这两本书列为推荐读物。这两本书覆盖了大学算法课程和数据结构课程的大部分内容，但是与普通教材的侧重点又不一樣，不强调单纯从数学上来进行分析的技巧，而是强调结合实际问题来进行分析、应用和实现的技巧，因此可作为大学计算机专业的算法、数据结构、软件工程等课程的教师参考用书和优秀课外读物。书中有许多真实的历史案例和许多极好的练习题以及部分练习题的提示与解答，非常适合自学。正如作者所建议的那样，阅读这两本书时，读者需要备有纸和笔，最好还有一台计算机在手边，边读边想、边想边做，这样才能将阅读这两本书的收益最大化。

人民邮电出版社引进版权，同时翻译出版了《编程珠玑（第2版）》和《编程珠玑（续）》，使这两个中译本珠联璧合，相信这不仅能极大地满足广大程序员读者的需求，还有助于提高国内相关课程的授课质量和学生的学习兴趣。

本书主要由黄倩和钱丽艳翻译，刘田审校，翻译过程中得到了张怀勇先生的帮助，在此表示感谢。由于本书内容深刻，语言精妙，而译者的水平和时间都比较有限，错误和不当之处在所难免，敬请广大读者批评指正。

## 前言

计算机编程有很多方面。Fred Brooks在《人月神话》一书中为我们描绘了全景，他的文章强调了管理在大型软件项目中所起的关键作用。而Steve McConnell在《代码大全》一书中更具体地传授了良好的编程风格。这两本书所讨论的是好软件的关键因素和专业程序员应有的特征。遗憾的是，仅仅熟练地运用这些可靠的工程原理，不见得一定能够如期完成软件并顺利运行。

关于本书

本书描述了计算机编程更具魅力的一面：在可靠的工程之外，在洞察力和创造力范围内结晶而出的编程珠玑。正如自然界中的珍珠来自于磨砺牡蛎的细沙一样，这些编程珠玑来自于磨砺程序员的实际问题。书中的程序都很有趣，传授了重要的编程技巧和基本的设计原理。

本书大部分内容最初发表在《ACM通讯》中我主持的“编程珠玑”专栏。这些内容经过汇总和修订，在1986年集结出版，成为了本书的第1版。第1版的13篇文章中，有12篇都在本版中做了大幅修订；此外，本版还补充了3篇新的内容。

阅读本书所需的唯一背景知识就是某种高级语言的编程经验。书中偶尔会出现一些高级技术（如C++中的模板等），对此不熟悉的读者可以跳过这些内容，基本上不影响阅读。

本书每一章都独立成篇，各章之间却又有逻辑分组。第1章至第5章构成本书的第一部分，这部分回顾了编程的基本原理：问题定义、算法、数据结构以及程序验证和测试。第二部分围绕效率这个主题展开。效率问题有时本身很重要，又永远都是进入有趣编程问题的绝佳跳板。第三部分用这些技术来解决排序、搜索和字符串等重要问题。

阅读本书的一个提示：不要读得太快。要仔细阅读，一次读一章。要尝试解答书中提出的问题——有些问题需要集中精力思考一两个小时才会变得容易。然后，要努力解答每章末尾的习题：当读者写下答案时，从本书学到的大部分知识就会跃然纸上。如有可能，要先与朋友和同事讨论一下自己的思路，再去查阅本书末尾的提示和答案。每章末尾的“深入阅读”并不算是学术意义上的参考文献表，而是我推荐的一些好书，这些书是我个人藏书的重要部分。

本书是为程序员而写的。我希望书中的习题、提示、答案和深入阅读对每个人都有用。本书已用作算法、程序验证和软件工程等课程的教材。附录A中的算法分类可供实际编程人员参考，该附录同时还说明了如何在算法和数据结构课程中使用本书。

代码

本书第1版中的伪代码程序其实都已实现，但当时未公开。在本版中，我重写了所有的老程序，并且编写了差不多等量的新代码。这些程序可以在<http://netlib.bell-labs.com/cm/cs/pearls/>下载。代码中包含许多对函数进行测试、调试和计时的脚手架程序。该网站还提供了其他相关的材料。由于现在许多的软件都能在线获得，因此本版的一个新增内容就是：如何评估和使用软件组件。

本书的程序采用了简洁的代码风格：短变量名，很少空行，很少或没有错误检测。这种风格不适用于大型软件项目，却有助于表达算法的核心思想。第5章第1个习题的答案给出了这种风格的更多细节。

本书包含几个实际的C和C++程序，其余大多数函数都用伪代码来表示，这样既节省了空间，又避免了繁琐的语法。记号for  $i = [0, n)$ 表示在从0至n-1的范围内对i进行迭代。在这类for循环中，左圆括号和右圆括号代表开区间（不包括端点值），而左方括号和右方括号代表闭区间（包括端点值）。表达式function(i,j)仍表示用参数i和j调用函数，而array[i,j]仍表示访问数组元素。

本版所提供的许多程序的运行时间都基于“我的计算机”——一台128 MB内存、运行Windows NT 4.0操作系统的400 MHz Pentium II。我测试了这些程序在其他几台机器上的运行时间，书中记录了我观察到的一些显著的差异。所有的实验都使用了最高级别的编译器优化。建议读者在自己的计算机上对这些程序计时，我敢打赌读者将会发现相似比率的运行时间。

致第1版的读者

我希望你们在翻阅本版时的第一感觉是“看起来很眼熟啊”，而过几分钟又得出结论“以前从来没读过”。

本版与第1版主题相同，但涉及的范围更广。计算技术已经在数据库、网络 and 用户界面等重要领域取得了长足的进展。大多数程序员应当都熟悉这些技术。但是，这些领域的中心仍然是那些核心编程问题，这些问题还是本书的主题。相对于第1版而言，本版可以比喻为一条稍微长大了的鱼，游进了一个大得多的池塘。

第1版第4章关于实现二分搜索的一节内容经过扩充成为了本版中关于测试、调试和计时的第5章。第1版第11章经过扩充，在本版中分成了第12章（还讨论原来的问题）和第13章（讨论集合表示）。第1版第13章描述的在64 KB地址空间运行的拼写检查器已被删除，但其要点仍保留在13.8节中。新增的第15章讨论字符串问题。本版在第1版的各章中插入了许多新节，同时删除了一些旧节。新增的习题、答案以及4个附录使得本版篇幅比第1版增加了25%。

本版保留了许多原有的实例研究，因为它们具有历史价值。有些老故事则用现代术语做了改写。

第1版的致谢

对许多人给予我的诸多帮助，我一直心存感激。Peter Denning和Stuart Lynn最早设想在《ACM通讯》上开设专栏。Peter在计算机学会（ACM）内做了大量的工作，促成了该专栏，并动员我来主持这个专栏。ACM总部员工（特别是Roz Steier和Nancy Adriance）在本书各篇文章最初发表时给予大力协助。我要特别感谢ACM鼓励我以目前这种经过修订的形式来出版各篇文章；还要特别感谢《ACM通讯》的众多读者，他们对原始各篇文章的评论使得这个扩充版本成为必要的和可能的。

Al Aho、Peter Denning、Mike Garey、David Johnson、Brian Kernighan、John Linderman、Doug McIlroy和Don Stanat都非常仔细地读过每一章，尽管时间期限常常很紧。我还要感谢以下诸位的宝贵意见：Henry Baird、Bill Cleveland、David Gries、Eric Grosse、Lynn Jelinski、Steve Johnson、Bob Melville、Bob Martin、Arno Penzias、Marilyn Roper、Chris Van Wyk、Vic Vyssotsky和Pamela Zave。Al Aho、Andrew Hume、Brian Kernighan、Ravi



Sethi、Laura Skinger和Bjarne Stroustrup在本书的成书过程中给予了无法估量的帮助，而西点军校EF 485课程的学员实际核对了倒数第二稿 [1]。再次谢谢诸位。

#### 第2版的致谢

Dan Bentley、Russ Cox、Brian Kernighan、Mark Kernighan、John Linderman、Steve McConnell、Doug McIlroy、Rob Pike、Howard Trickey和Chris Van Wyk都非常仔细地阅读过本版。我还要感谢以下诸位的宝贵意见：Paul Abrahams、Glenda Childress、Eric Grosse、Ann Martin、Peter McIlroy、Peter Memishian、Sundar Narasimhan、Lisa Ricker、Dennis Ritchie、Ravi Sethi、Carol Smith、Tom Szymanski和Kentaro Toyama。感谢Addison-Wesley出版社的Peter Gordon和他的同事们帮助筹划了本版。

Jon Bentley

于新泽西州Murray Hill

1985年12月

1999年8月

---

[1] 原作者在给译者的电子邮件中指出，他曾在西点军校授课，用本书草稿作为教材，EF为Engineering Fundamentals（工程基础）系的缩写。  
——译者注

# 第一部分 基础

这一部分的5章回顾程序设计的基础知识。第1章介绍一个问题的历史，我们把仔细的问题定义和直接的程序设计技术结合起来，得到优美的解决方案。这一章揭示了本书的中心思想：对实例研究的深入思考不仅很有趣，而且可以获得实际的益处。

第2章讨论3个问题，其中重点强调了如何由算法的融会贯通获得简单而高效的代码。第3章总结数据结构在软件设计中所起到的关键作用。

第4章介绍一个编写正确代码的工具——程序验证。在第9章、第11章和第14章中生成复杂（且快速）的函数时，大量使用了程序验证技术。第5章讲述如何把这些抽象的程序变成实际代码：使用脚手架程序来探测函数，用测试用例来测试函数并度量函数的性能。

本部分内容

第1章 开篇

第2章 啊哈！算法

第3章 数据决定程序结构

第4章 编写正确的程序

第5章 编程小事

## 第1章 开篇

一位程序员曾问我一个很简单的问题：“怎样给一个磁盘文件排序？”想当年我是一上来就犯了错误，现在，在讲这个故事之前，先给大家一个机会，看看能否比我当年做得更好。你会怎样回答上述问题呢？

### 1.1 一次友好的对话

我错就错在马上回答了这个问题。我告诉他一些有关如何在磁盘上实现归并排序的简要思路。我建议他深入研究算法教材，他似乎不太感冒。他更关心如何解决这个问题，而不是深入学习。于是我告诉他在一本流行的程序设计书里有磁盘排序的程序。那个程序有大约200行代码和十几个函数，我估计他最多需要一周时间来实现和测试该代码。

我以为已经解决了他的问题，但是他的踌躇使我返回到了正确的轨道上。其后就有了下面的对话，楷体部分是我的问题。

为什么非要自己编写排序程序呢？为什么不用系统提供的排序功能呢？

我需要在一个大系统中排序。由于不明的技术原因，我不能使用系统中的文件排序程序。

需要排序的内容是什么？文件中有多少条记录？每条记录的格式是什么？

文件最多包含1千万条记录，每条记录都是7位的整数。

等一下，既然文件这么小，何必非要在磁盘上进行排序呢？为什么不在内存里进行排序呢？

尽管机器有许多兆字节的内存，但排序功能只是大系统中的一部分，所以，估计到时只有1 MB的内存可用。

你还能告诉我其他一些与记录相关的信息吗？

每条记录都是7位的正整数，再无其他相关数据。每个整数最多只出现一次。

这番对话让问题更明确了。在美国，电话号码由3位“区号”后再跟7位数字组成。拨打含“免费”区号800（当时只有这一个号码）的电话是不收费的。实际的免费电话号码数据库包含大量的信息：免费电话号码、呼叫实际中转到的号码（有时是几个号码，这时需要一些规则来决定哪些呼叫在什么时间中转到哪里）、主叫用户的姓名和地址等。

这位程序员正在开发这类数据库的处理系统的一小部分，需要排序的整数就是免费电话号码。输入文件是电话号码的列表（已删除所有其他信息），号码重复出现算出错。期望的输出文件是以升序排列的电话号码列表。应用背景同时定义了相应的性能需求。当与系统的会话时间较长时，用户大约每小时请求一次有序文件，并且在排序未完成之前什么都干不了。因此，排序最多只允许执行几分钟，10秒钟是比较理想的运行时间。

### 1.2 准确的问题描述

对程序员来说，这些需求加起来就是：“如何给磁盘文件排序？”在试图解决这个问题之前，先将已知条件组织成一种更客观、更易用的形式。

输入：一个最多包含  $n$  个正整数的文件，每个数都小于  $n$ ，其中  $n=10^7$ 。如果在输入文件中有任何整数重复出现就是致命错误。没有其他数据与该整数相关联。

输出：按升序排列的输入整数的列表。

约束：最多有（大约）1 MB的内存空间可用，有充足的磁盘存储空间可用。运行时间最多几分钟，运行时间为10秒就不需要进一步优化了。

请花上一分钟思考一下该问题的规范说明。现在你打算给程序员什么样的建议呢？

### 1.3 程序设计

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《编程珠玑（第2版·修订版）》[美]乔恩·本特利（Jon Bentley）著.epub

请登录 <https://shgis.cn/post/327.html> 下载完整文档。

手机端请扫码查看：

