

TURING

图灵原版计算机科学系列

PEARSON

计算机程序设计艺术

卷1：基本算法

(英文版·第3版)

[美] Donald E. Knuth 著

 人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵原版计算机科学系列

计算机程序设计艺术

卷1：基本算法

(英文版·第3版)

The Art of Computer Programming
Vol 1: Fundamental Algorithms
Third Edition

人民邮电出版社
北京

图书在版编目 (CIP) 数据

计算机程序设计艺术：第3版. 第1卷, 基本算法：
英文 / (美) 高德纳 (Knuth, D. E.) 著. — 北京：人民
邮电出版社, 2010. 10

(图灵原版计算机科学系列)

书名原文: The Art of Computer Programming, Vol
1: Fundamental Algorithms, Third Edition
ISBN 978-7-115-23259-5

I. ①计… II. ①高… III. ①程序设计—英文 IV.
①TP311.1

中国版本图书馆CIP数据核字(2010)第122114号

内 容 提 要

《计算机程序设计艺术》系列被公认为计算机科学领域的权威之作, 深入阐述了程序设计理论, 对计算机领域的发展有着极为深远的影响。本书是该系列的第1卷, 讲解基本算法, 其中包含了其他各卷都需用到的基本内容。本卷从基本概念开始, 然后讲述信息结构, 并辅以大量的习题及答案。

本书适合从事计算机科学、计算数学等各方面工作的人员阅读, 也适合高等院校相关专业的师生作为教学参考书, 对于想深入理解计算机算法的读者, 是一份必不可少的珍品。

图灵原版计算机科学系列

计算机程序设计艺术 卷1: 基本算法 (英文版·第3版)

◆ 著 [美] Donald E. Knuth

责任编辑 杨海玲

执行编辑 谢灵芝

◆ 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 电子函件 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: 700×1000 1/16

印张: 42

字数: 807千字 2010年10月第1版

印数: 1-2 500册 2010年10月北京第1次印刷

著作权合同登记号 图字: 01-2010-4226号

ISBN 978-7-115-23259-5

定价: 119.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original edition, entitled *The Art of Computer Programming, Vol 1: Fundamental Algorithms, Third Edition*, 9780201896831 by Donald E. Knuth, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 1997 by Pearson Education, Inc.

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.

China edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2010.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in the People's Republic of China excluding Hong Kong, Macao and Taiwan.

本书英文版由 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。

仅限于中华人民共和国境内（香港、澳门特别行政区和台湾地区除外）销售发行。

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。

*This series of books is affectionately dedicated
to the Type 650 computer once installed at
Case Institute of Technology,
in remembrance of many pleasant evenings.*

前 言

本书是应数以千计的读者来信要求而出版的。
我们用了多年的时间对大量的食谱进行了反复检验，
挑选出最佳的、有趣的、完美的食谱奉献给大家。

现在我们可以自信满满地说，
不管是谁，即使此前从来没有做过菜，
只要严格按书中的说明进行操作，也能获得跟我们一样的烹饪效果。

—— *McCall's Cookbook* (1963)

为数字计算机编写程序的过程是特别吸引人的，因为我们不仅可以获得经济和科学两方面的收益，还能尽享写诗或作曲般的艺术体验。本书是多卷本中的第1卷，整套书旨在训练读者去掌握程序员必备的各种技能。

在接下来的章节中，我不打算介绍计算机程序设计的入门知识，而是假定读者已有一定的基础。必备知识实际上非常简单，但初学者恐怕需要一些时间和动手实践方能理解数字计算机的概念。读者应该具备如下知识。

a) 对存储程序式数字计算机的工作原理有一些认识。不一定需要电子学背景，但需要知道指令在机器内存中是如何保存和连续执行的。

b) 能够用计算机可以“理解”的确切术语来描述问题的解决方案。（这些机器不懂所谓的常识，它们只会精准地按要求干活，不会多做也不会少做。这是刚开始接触计算机时最难领悟的概念。）

c) 掌握一些最基本的计算机技术，如循环（重复地执行一组指令）、子程序的使用、下标变量的使用。

d) 对常见的计算机术语有所了解，如内存、寄存器、位、浮点、溢出、软件等。正文中没有下定义的一些术语，会在每卷最后的索引部分给出简明的定义。

或许可以把这4点归结为一个要求：读者应该起码为一台计算机编写和测试过至少（比如说）4个程序。

我力图使这套书能满足两方面的需求。首先，这些书总结了几个重要领域的知识，可以作为参考书；其次，它们可以用作自学教材或计算机与信息科学专业的大学教材。为此，我在书中加入了大量的习题，并为多数习题提供了答案。此外，我

在书中尽可能地用事实说话，言之有据，避免做一些含糊的泛泛而谈。

这套书针对的读者不是那些对计算机只有一时兴趣的人，但也决不仅限于计算机专业人士。其实，我的一个主要目标是使其他领域的广大工作者能够方便地了解这些程序设计方法，他们本可以利用计算机获得更丰硕的成果，但却没时间去技术刊物中查找必需的信息。

这套书的主题可以称为“非数值分析”。在传统意义上，计算机是与方程求根、数值插值与积分等数值问题求解联系在一起的，但我不会去专门讨论数值问题，最多只顺带提一下。数值计算的程序设计是一个发展迅速、引人入胜的领域，目前已出版了许多这方面的书。不过，从20世纪60年代初期开始，计算机更多地是用于处理那些很少出现数值的问题，此时用到的是计算机的决策能力而非算术运算能力。非数值问题中也会用到一些加法和减法，但基本不需要乘法和除法。当然，即便是主要关注数值计算程序设计的人也可以在非数值方法的学习中受益，因为数值程序中也会用到非数值方法。

非数值分析的研究成果散见于大量科技刊物中。通过研究，我从大量文献中提炼出了一些可以用于多种编程场合的最基本的方法，并尝试着将其组织为一套“理论”。同时，我还展示了如何应用这一理论解决大量的实际问题。

当然，“非数值分析”是一种太过否定性的提法，使用肯定性的描述语来刻画这一主题要好得多。“信息处理”对于我们的内容而言过于宽泛，“程序设计技术”又显得太狭窄了，因此我提出用算法分析来概括这套书的主题比较恰当，这一名称暗示我们将探讨“与特定计算机算法的性质有关的理论”。

整套书以“计算机程序设计艺术”为题，内容总览如下。

- 第1卷 基础算法
 - 第1章 基本概念
 - 第2章 信息结构
- 第2卷 半数值算法
 - 第3章 随机数
 - 第4章 算术
- 第3卷 排序与查找
 - 第5章 排序
 - 第6章 查找
- 第4卷 组合算法
 - 第7章 组合查找
 - 第8章 递归
- 第5卷 语法算法
 - 第9章 词法扫描
 - 第10章 语法分析

第4卷涉及的范围很广，实际上是3本书（卷4A、4B和4C）。此外，我还计划编写两卷更具专业性的内容，即第6卷《语言理论》（第11章）和第7卷《编译器》（第12章）。

1962年，我打算按顺序把上述各章组织成一本书，但很快就意识到应该深入介绍这些主题，而不能蜻蜓点水般地只讲些皮毛。最终写成的篇幅表明，每一章包含的内容都无法在一学期的大学课程中讲完，因此只能分卷出版。我知道一本书仅有一章或两章会显得很奇怪，但为了便于相互参照我决定保留原来的章节编号。我计划出一本第1卷至第5卷的精简版，供大学用作本科计算机课程的参考书或教材。该精简版的内容是这5卷的子集，但略去了专业性较强的信息。精简版与完整版的章节编号将保持一致。

本卷可以看成是整套书的“交集”，因为它包含了其他各卷都需要用到的基本内容，而第2卷至第5卷则可以彼此独立地阅读。第1卷不仅可以作为阅读其他各卷的参考书，还可以用作数据结构（主要是第2章的内容）、离散数学（主要是1.1节、1.2节、1.3.3节和2.3.4节的内容）、机器语言程序设计（主要是1.3节和1.4节的内容）等课程的大学教材或自学读物。

我撰写本书的出发点不同于当前的大多数计算机程序设计书籍。我教给读者的不是如何使用别人的软件，而是如何自己编写更好的软件。

我最初的目标是引领读者到达每个主题的知识前沿。然而，身处一个经济上有利可图的领域，要保持不落伍是极端困难的，计算机科学的迅猛发展使我这一梦想最终破灭了。我们讨论的每一个主题都包含了全世界成千上万的能人异士所贡献出的成千上万的精妙成果。因此，我一改初衷，开始专注于那些很可能在多年后仍然很重要的“经典”方法，并尽我所能来描述它们。特别是，我尽力跟踪了每个主题的演变历史，为其未来的发展打好坚实的基础；尽力选择了与当前用法一致的简明术语；并尽力囊括了与顺序计算机程序设计相关的所有美妙而易于表述的思想。

对于这套书的数学内容还需要说几句。从内容选择上来说，具有高中代数知识的普通读者就可以阅读这些书，遇到涉及较多数学知识的部分时可以略读，而偏好数学的读者则可以学到许多与离散数学相关的很有意思的数学方法。在行文上为兼顾这两类读者，我给每道习题都标定了等级，数学性强的题目会被特别标示出来，我还在多数章节里把主要数学结果放在其证明之前陈述。相关证明要么留作习题（答案单独用一节集中给出），要么在小节的最后给出。

如果读者主要对程序设计而非相关的数学内容感兴趣，那么可以在感觉数学难度显著加大时停止阅读这一节，而对数学有兴趣的读者则能在这些节发现许多丰富而有趣的内容。不少已经发表的与计算机程序设计有关的数学文章都含有错误，本书的目标之一就是正确的数学方法向读者传授相关的内容。我自认为是一名数学家，因此会竭尽所能地维护数学的完整性，对此我责无旁贷。

其实有基本的微积分知识就足够理解这套书中的多数数学内容，因为所需的其他理论大多是由此建立起来的。当然，有时我会用到复变函数理论、概率论、数论等的更深入的定理，这种情况下我会指出相应的教科书。

撰写这套书时最艰难的抉择在于如何展示各种各样的方法。采用流程图和算法的非形式化的逐步描述显然很有优势，相关讨论可以参考《ACM 通讯》第 6 卷（1963 年 9 月）第 555~563 页的“Computer-Drawn Flowcharts”（计算机绘制流程图）一文。不过，描述任何计算机算法时，形式化的准确语言也是必不可少的，我需要决定是使用 ALGOL 或 FORTRAN 这样的代数语言，还是使用一种面向机器的语言。可能当今的许多计算机专家都不赞同我最后使用了面向机器的语言，但我确信自己的选择是恰当的，理由如下。

a) 程序员在很大程度上受编程语言的影响。如今的主流趋势就是，使用语言中最简单的结构而不是对机器而言性能最好的特性来构建程序。如果学习并理解了面向机器的语言，程序员会倾向于使用一种效率高得多的方法，这更贴近实际。

b) 除了个别情况，我们所用的程序都相当短小。因此只要有台合适的计算机，理解这些程序不会有什么难度。

c) 高级语言不适合用来讨论协同程序链接、随机数生成、多精度算术以及与内存有效使用相关的许多问题的底层细节。

d) 只要不是对计算机只有一时兴趣的人，都应该学好机器语言，这是计算机的基础。

e) 无论如何，我们都需要某种机器语言作为许多示例中的软件程序的输出。

f) 代数语言每隔 5 年左右就会更新换代，而我想强调的是永恒的思想。

从另一个角度来看，我承认如果使用高级程序设计语言，程序的编写要容易一些，程序的调试更是容易很多。事实上，计算机已非常强大快速，我从 1970 年开始就很少使用低级的机器语言编写程序了。然而，对于书中的许多有趣的问题而言，程序员的技巧是至关重要的。例如，某些组合计算需要重复 1 万亿次，其内循环时间每减少 1 微秒，我们就能节省 11.6 天。同样，多花些时间编写每天需要在许多计算机装置中多次使用的软件也是值得的，因为软件只需要编写一次。

既然已经决定了要使用面向机器的语言，下一个问题就是应该用哪种语言了。我可以选择特定机器 X 上的语言，但这样的话，没有机器 X 的人就会认为本书是仅仅针对 X 用户的。此外，机器 X 可能具有许多与本书内容无关的特性，但仍得加以解释；两年后，生产机器 X 的厂家可能会推出机器 $X + 1$ 或机器 $10X$ ，那时候就不会再有人对机器 X 感兴趣了。

为了避免陷入这样的窘境，我尝试着设计了一种操作规则非常简单（甚至只要一个小时就能掌握）的“理想”计算机，它跟实际的机器很类似。学生没有理由对学习多种计算机的特性感到恐惧：一旦掌握了某种机器语言，其他机器的语言自然就很简单了。毫无疑问，真正的程序员在职业生涯中会遇到许多种机器语言。对于

我的假想机而言，现在唯一的缺陷就是难以执行为它编写的程序。幸运的是，许多志愿者已经自告奋勇地站出来为其编写模拟程序了，因此程序的执行也不再是问题了。这些模拟程序非常适合于教学目的，因为它们甚至比实际的计算机还要易于使用。

我尽量在讨论每个主题时引用最好的早期论文，并摘录一些新近的进展。对期刊文献的引用一般采用标准的缩写方式，但下列引用最频繁的杂志例外，它们的缩写如下：

CACM = Communications of the Association for Computing Machinery (《ACM 通讯》)

JACM = Journal of the Association for Computing Machinery (《ACM 杂志》)

COMP. J. = The Computer Journal (British Computer Society) (英国计算机学会《计算机杂志》)

Math. Comp. = Mathematics of Computation (《计算数学》)

AMM = American Mathematical Monthly (《美国数学月刊》)

SICOMP = SIAM Journal on Computing (《SIAM 计算杂志》)

FOCS = IEEE Symposium on Foundations of Computer Science (《IEEE 计算机科学基础论文集》)

SODA = ACM-SIAM Symposium on Discrete Algorithms (《ACM-SIAM 离散算法论文集》)

STOC = ACM Symposium on Theory of Computing (《ACM 计算理论论文集》)

Crelle = Journal für die reine und angewandte Mathematik (《理论与应用数学杂志》)

例如，我将用“*CACM* 6 (1963), 555-563”表示前面提到的那篇文献。此外，我用“*CMath*”表示 *Concrete Mathematics* (《具体数学》) 一书，1.2 节的引言部分会引用到它。

这套书的许多技术性内容出现在习题中。有些重要习题的构思不是我的原创，我会尽力给出原创者的信息。相应的文献引用通常在该节的正文或者习题的答案中给出。但许多情况下习题是基于未发表的材料，此时就无法给出进一步的文献信息了。


许多人在这套书的撰写过程中给予过我帮助，在此表示深深的谢意。首先是我的妻子 Jill，感谢她的极大耐心，感谢她为本书绘制了部分插图，感谢她各方面无尽的支持。其次感谢 Robert W. Floyd，他早在 20 世纪 60 年代就为增强这套书的内容投入了大量的时间。此外，数以千计的其他人士也提供了重要的帮助，如果要一一列出他们的名字还需要一本書的篇幅！他们中的许多人无私地允许我使用其尚未公开发表的内容。我在加州理工大学和斯坦福大学从事研究工作时，美国国家自然科学基金会和海军研究局给予了多年的慷慨赞助。从 1962 年我开始写这套书的时候

起, Addison-Wesley 公司就一直给予我极大的帮助与配合。我知道, 向他们表示感谢的最好方法就是尽我所能把书写成他们所期望的样子, 不负他们的投入。

第 3 版前言

花 10 年时间开发了用于计算机排版的 $\text{T}_{\text{E}}\text{X}$ 系统和 METAFONT 系统之后, 我现在已经能够实现当初的一个梦想了: 用这些系统来排版《计算机程序设计艺术》。最终, 本书的全部内容都存储在我的个人电脑中, 该电子格式可以随时适应印刷技术和显示技术的改进。新的设置使得我可以在文字上进行数千处的改进, 这是我多年来一直想做的。

我对新版的文字逐字进行了认真的审阅, 力图在保持原有的蓬勃朝气的同时加入一些可能更成熟的论断。新版增加了几十道新的习题, 并为原有的几十道习题给出了改进过的新答案。

 《计算机程序设计艺术》这套书尚未完稿, 因此书中有些部分还带有“建设中”的图标, 以向读者致歉——这部分内容还不是最新的。我电脑中的文件里堆满了重要的材料, 打算写进第 1 卷最终的壮丽无比的第 4 版中, 或许从现在算起还需要 15 年的时间。但我必须先完成第 4 卷和第 5 卷, 非到万不得已, 我不想拖延这两卷的出版时间。

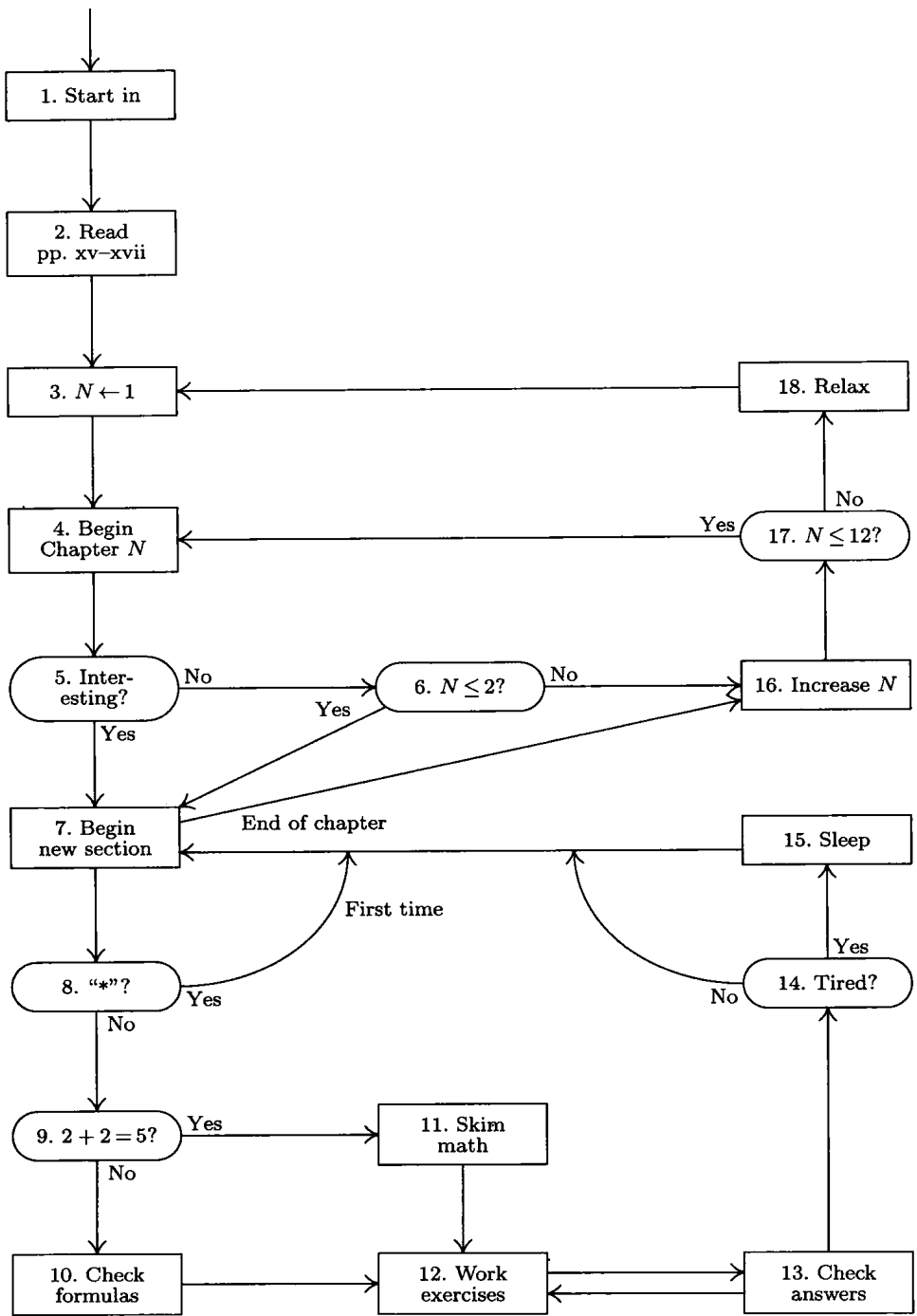
准备这一新版本的大部分艰苦工作是由 Phyllis Winkler、Silvio Levy 和 Jeffrey Oldham 完成的。Phyllis 和 Silvio 把第 2 版的全部文本熟练地录入了计算机, 并进行了编辑, Jeffrey 则几乎把所有的插图都转换成了 METAPOST 格式。我修正了细心的读者在第 2 版中发现的所有错误 (以及读者未能察觉的一些错误), 并尽量避免在新增内容中引入新的错误。但我想可能仍然有一些缺陷, 我希望能尽快加以改正。因此, 我很乐意向首先发现技术性错误、印刷错误或历史知识错误的人按每处错误 2.56 美元支付报酬。下列网页上列出了所有已反馈给我的错误的最新更正: <http://www-cs-faculty.stanford.edu/~knuth/taocp.html>。

高德纳 (Donald E. Knuth)

1997 年 4 月于加利福尼亚州斯坦福市

过去 20 年里发生了巨变。

——比尔·盖茨 (1995)



Flow chart for reading this set of books.

Procedure for Reading This Set of Books

1. Begin reading this procedure, unless you have already begun to read it. *Continue to follow the steps faithfully.* (The general form of this procedure and its accompanying flow chart will be used throughout this book.)
2. Read the Notes on the Exercises, on pages xv–xvii.
3. Set N equal to 1.
4. Begin reading Chapter N . Do *not* read the quotations that appear at the beginning of the chapter.
5. Is the subject of the chapter interesting to you? If so, go to step 7; if not, go to step 6.
6. Is $N \leq 2$? If not, go to step 16; if so, scan through the chapter anyway. (Chapters 1 and 2 contain important introductory material and also a review of basic programming techniques. You should at least skim over the sections on notation and about MIX.)
7. Begin reading the next section of the chapter; if you have already reached the end of the chapter, however, go to step 16.
8. Is section number marked with “*”? If so, you may omit this section on first reading (it covers a rather specialized topic that is interesting but not essential); go back to step 7.
9. Are you mathematically inclined? If math is all Greek to you, go to step 11; otherwise proceed to step 10.
10. Check the mathematical derivations made in this section (and report errors to the author). Go to step 12.
11. If the current section is full of mathematical computations, you had better omit reading the derivations. However, you should become familiar with the basic results of the section; they are usually stated near the beginning, or in *slanted type* right at the very end of the hard parts.
12. Work the recommended exercises in this section in accordance with the hints given in the Notes on the Exercises (which you read in step 2).
13. After you have worked on the exercises to your satisfaction, check your answers with the answer printed in the corresponding answer section at the

rear of the book (if any answer appears for that problem). Also read the answers to the exercises you did not have time to work. *Note:* In most cases it is reasonable to read the answer to exercise n before working on exercise $n + 1$, so steps 12–13 are usually done simultaneously.

14. Are you tired? If not, go back to step 7.
15. Go to sleep. Then, wake up, and go back to step 7.
16. Increase N by one. If $N = 3, 5, 7, 9, 11,$ or 12 , begin the next volume of this set of books.
17. If N is less than or equal to 12 , go back to step 4.
18. Congratulations. Now try to get your friends to purchase a copy of Volume 1 and to start reading it. Also, go back to step 3.

Woe be to him that reads but one book.

— GEORGE HERBERT, *Jacula Prudentum*, 1144 (1640)

*Le défaut unique de tous les ouvrages
c'est d'être trop longs.*

— VAUVENARGUES, *Réflexions*, 628 (1746)

Books are a triviality. Life alone is great.

— THOMAS CARLYLE, *Journal* (1839)

NOTES ON THE EXERCISES

THE EXERCISES in this set of books have been designed for self-study as well as classroom study. It is difficult, if not impossible, for anyone to learn a subject purely by reading about it, without applying the information to specific problems and thereby being encouraged to think about what has been read. Furthermore, we all learn best the things that we have discovered for ourselves. Therefore the exercises form a major part of this work; a definite attempt has been made to keep them as informative as possible and to select problems that are enjoyable as well as instructive.

In many books, easy exercises are found mixed randomly among extremely difficult ones. This is sometimes unfortunate because readers like to know in advance how long a problem ought to take—otherwise they may just skip over all the problems. A classic example of such a situation is the book *Dynamic Programming* by Richard Bellman; this is an important, pioneering work in which a group of problems is collected together at the end of some chapters under the heading “Exercises and Research Problems,” with extremely trivial questions appearing in the midst of deep, unsolved problems. It is rumored that someone once asked Dr. Bellman how to tell the exercises apart from the research problems, and he replied, “If you can solve it, it is an exercise; otherwise it’s a research problem.”

Good arguments can be made for including both research problems and very easy exercises in a book of this kind; therefore, to save the reader from the possible dilemma of determining which are which, *rating numbers* have been provided to indicate the level of difficulty. These numbers have the following general significance:

Rating Interpretation

- 00 An extremely easy exercise that can be answered immediately if the material of the text has been understood; such an exercise can almost always be worked “in your head.”
- 10 A simple problem that makes you think over the material just read, but is by no means difficult. You should be able to do this in one minute at most; pencil and paper may be useful in obtaining the solution.
- 20 An average problem that tests basic understanding of the text material, but you may need about fifteen or twenty minutes to answer it completely.

- 30 A problem of moderate difficulty and/or complexity; this one may involve more than two hours' work to solve satisfactorily, or even more if the TV is on.
- 40 Quite a difficult or lengthy problem that would be suitable for a term project in classroom situations. A student should be able to solve the problem in a reasonable amount of time, but the solution is not trivial.
- 50 A research problem that has not yet been solved satisfactorily, as far as the author knew at the time of writing, although many people have tried. If you have found an answer to such a problem, you ought to write it up for publication; furthermore, the author of this book would appreciate hearing about the solution as soon as possible (provided that it is correct).

By interpolation in this "logarithmic" scale, the significance of other rating numbers becomes clear. For example, a rating of 17 would indicate an exercise that is a bit simpler than average. Problems with a rating of 50 that are subsequently solved by some reader may appear with a 45 rating in later editions of the book, and in the errata posted on the Internet.

The remainder of the rating number divided by 5 indicates the amount of detailed work required. Thus, an exercise rated 24 may take longer to solve than an exercise that is rated 25, but the latter will require more creativity.

The author has tried earnestly to assign accurate rating numbers, but it is difficult for the person who makes up a problem to know just how formidable it will be for someone else to find a solution; and everyone has more aptitude for certain types of problems than for others. It is hoped that the rating numbers represent a good guess at the level of difficulty, but they should be taken as general guidelines, not as absolute indicators.

This book has been written for readers with varying degrees of mathematical training and sophistication; as a result, some of the exercises are intended only for the use of more mathematically inclined readers. The rating is preceded by an *M* if the exercise involves mathematical concepts or motivation to a greater extent than necessary for someone who is primarily interested only in programming the algorithms themselves. An exercise is marked with the letters "*HM*" if its solution necessarily involves a knowledge of calculus or other higher mathematics not developed in this book. An "*HM*" designation does *not* necessarily imply difficulty.

Some exercises are preceded by an arrowhead, "►"; this designates problems that are especially instructive and especially recommended. Of course, no reader/student is expected to work *all* of the exercises, so those that seem to be the most valuable have been singled out. (This is not meant to detract from the other exercises!) Each reader should at least make an attempt to solve all of the problems whose rating is 10 or less; and the arrows may help to indicate which of the problems with a higher rating should be given priority.

Solutions to most of the exercises appear in the answer section. Please use them wisely; do not turn to the answer until you have made a genuine effort to

solve the problem by yourself, or unless you absolutely do not have time to work this particular problem. *After* getting your own solution or giving the problem a decent try, you may find the answer instructive and helpful. The solution given will often be quite short, and it will sketch the details under the assumption that you have earnestly tried to solve it by your own means first. Sometimes the solution gives less information than was asked; often it gives more. It is quite possible that you may have a better answer than the one published here, or you may have found an error in the published solution; in such a case, the author will be pleased to know the details. Later editions of this book will give the improved solutions together with the solver's name where appropriate.

When working an exercise you may generally use the answers to previous exercises, unless specifically forbidden from doing so. The rating numbers have been assigned with this in mind; thus it is possible for exercise $n + 1$ to have a lower rating than exercise n , even though it includes the result of exercise n as a special case.

Summary of codes:	00	Immediate
	10	Simple (one minute)
	20	Medium (quarter hour)
▶ Recommended	30	Moderately hard
<i>M</i> Mathematically oriented	40	Term project
<i>HM</i> Requiring "higher math"	50	Research problem

EXERCISES

- ▶ 1. [00] What does the rating "*M20*" mean?
2. [10] Of what value can the exercises in a textbook be to the reader?
3. [14] Prove that $13^3 = 2197$. Generalize your answer. [This is an example of a horrible kind of problem that the author has tried to avoid.]
4. [*HM45*] Prove that when n is an integer, $n > 2$, the equation $x^n + y^n = z^n$ has no solution in positive integers x, y, z .

*We can face our problem.
We can arrange such facts as we have
with order and method.*

— HERCULE POIROT, in *Murder on the Orient Express* (1934)