

经典计算机科学著作最新修订版

计算机程序设计艺术  
第1卷 基本算法  
(第3版)

The Art of Computer  
Programming

苏运霖 译

[美] DONALD E. KNUTH 著

  
Addison  
Wesley

国防工业出版社  
National Defence Industry Press  
<http://www.ndip.com.cn>

565

经典计算机科学著作最新修订版

711.1

K401

# 计算机程序设计艺术

第1卷

## 基本算法

(第3版)

[美] Donald E. Knuth 著  
苏运霖 译

国防工业出版社

·北京·

著作权合同登记号 图字:军-2001-018号

### 图书在版编目(CIP)数据

计算机程序设计艺术.第1卷,基本算法:第3版/  
(美)克努特(Knuth, D. E.)著;苏运霖译.—北京:  
国防工业出版社,2002.9

书名原文:The Art of Computer Programming

ISBN 7-118-02799-5

I. 计... II. ①克...②苏... III. 电子计算机-算  
法设计 IV. TP311

中国版本图书馆 CIP 数据核字(2002)第 007597 号

互联网网页 <http://www-cs-faculty.stanford.edu/~knuth/taocp.html> 包含本书及相关著作的最新信息。

Simplified Chinese edition copyright © 2002 by Pearson Education North Asia Limited and National Defense Industry Press.

Original English language title: The Art of Computer Programming, 3rd Edition by Donald E. Knuth

Copyright © 1997 by Addison Wesley Longman

All Rights Reserved.

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

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

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

开本 787×960 1/16 印张 40½ 880 千字  
2002 年 9 月第 1 版 2002 年 9 月北京第 1 次印刷  
印数:1-4000 册 定价:98.00 元

(本书如有印装错误,我社负责调换)

## 中文版前言(附原文)

谨向拥有此书的全体中国读者问候!我诚挚地向中国的程序员推荐我的中文名字“高德纳”,那是1977年在我访问你们的国家之前,由姚储枫(Francis Yao)为我起的。那次为期三周的访问至今仍给我留下美好的回忆。我高兴地看到自1989年以来,在《计算机科学与技术》杂志的刊头上有“高德纳”的名字。虽然我不通晓你们的语言,但这个名字已使我对全体中国人民倍感亲近。

献身于计算机程序设计的人们必须从事大量艰苦的工作,必须掌握许多精微的技术细节,因而在这方面深有造诣者不多,但其回报却是巨大的,因为一个编写得很好的程序完全可以成为一件美丽的艺术品,而且还因为,正是计算机程序使得全世界的人们彼此靠得更近。

Donald E. Knuth (高德纳)

## Preface to the Chinese Edition

Greetings to all readers of these books in China! I fondly hope that many Chinese computer programmers will learn to recognize my Chinese name “高德纳”, which was given to me by Francis Yao just before I visited your country in 1977. I still have very fond memories of that three-week visit, and I have been glad to see “高德纳” on the masthead of the *Journal of Computer Science and Technology* since 1989. This name makes me feel close to all Chinese people although I cannot speak your language.

People who devote much of their lives to computer programming must do a great deal of hard work and must master many subtle technical details. Not many are able to do this well. But the rewards are great, because a well written program can be a beautiful work of art, and because computer programs are helping to bring all people of the world closer together.

Donald E. Knuth (高德纳)

## 译者前言

在新千年到来的前夕,我在译稿上写下最后一个句号。这意味着几百万字的重译工作至此告一段落。我顿觉如释重负,浑身轻松;却又思潮澎湃,心绪激动。端笔记下,遂成此前言,留给己与人。

二十多年前,当中国大地还笼罩在“知识分子臭老九”的气氛下时,我在书店中偶然发现 D.E. 克努特(高德纳)这套传世之作,立即预感到它的巨大精神力量,遂萌生把它们译出,以供我国同仁共同从中获益的想法。当我把这个想法讲给当时和我一起共事、曾经是我老师的管纪文先生时,他也欣然同意,并愿意和我一起完成这件既苦又累、但又是很有意义的工作。

当年,当我和管纪文先生完成了全书的翻译工作时,我也曾有过这样一种兴奋激动的心情,今天,我似乎是在重新经历那时的感觉。

也许每一个曾经经历过那个时期的人们都会为二十多年来沧海桑田般的变化而感慨万千。当年由于众所周知的原因和社会气氛,为了出版这套书,实在是颇费周折的。因此我们要在这里感谢当年为使我们的译著得以出版的许多领导和专家,正是他们的真知灼见和无私无畏的精神,以及他们不畏艰难的支持和帮助,才使我们的工作免于“胎死腹中”。二十多年来的改革开放,确实为我们国家、为我们的人民带来了巨大变化。正是在这样的大气候下,才有我们这项工作的继续和这套书在中国的再版。

我深深地为作者锲而不舍、精益求精的精神所感动。作者在这套书第 1 版的前言中,曾经提到了他的十分复杂的感情——他说,从一方面说,他当然希望自己的书能永远地有价值,永远给人们提供教益;但另一方面,他也知道,对于迅猛发展的像计算机这样的学科,知识的更新是如此迅速,要想使自己的书经久不衰,永不过时,是做不到的。所以他只想使自己的书不至于或不要过时太快。然而,十分可贵的是,他没有停留在这种矛盾中,而是以一种不屈不挠的精神来延续自己著作的寿命和价值。为了做到这一点,他花费了大量的精力,投入了巨大的劳动对全书进行修订;最重要的是以当前这个领域的知识前沿来全面更新全书。他还不满足于此,他还要把书中原来使用的样板机 MIX 及其语言,更新成为新一代 MMIX。在作者身上,精益求精的精神体现得十分生动:对于他所研制的 T<sub>E</sub>X 和 METAFont 是这样——可以说他是以十年铸一剑来完成这两个誉满全球的计算机排版印刷的精品;对于他的科学研究是这样——可以说他总是要把每个所研究的问题都彻底解决了才甘罢休的;对于他的这套成名著作也是如此——他不但要使其内容经得起考验,也要让全书绝无错误。因此现在他又承诺以 2.56 美元的奖励来酬谢任何一个错误的头一个发现者。这种对工作高度认真负责,一丝不苟的精神,难道不值得我们为之赞叹,不值得我们立为楷模吗?

二十多年过去了,我和管纪文先生共同翻译的第1版离开书店已经很久了。在一些图书馆里,即使能找到,大概也早已皮破纸黄了。然而每当人们谈起它们时,仍然会以赞扬的口吻说,那几本书翻译得很好,对我国计算机的发展做出了很大贡献。我要感谢读者和同行们的鼓励。今天,我所期盼的仍然是,不仅得到读者和专家们对我付出的艰苦劳动的肯定,而且更希望我付出的这一劳动,对于我国的计算机界——特别是年轻一代的计算机界,产生积极的影响。我盼望他们认认真真地去阅读这套书(而且还不是为了作为找比尔·盖茨的敲门砖,尽管他说过,任何一个读懂这套书的人,都将得到他的赏识——他可以接受你的简历),以便为献身于我国自己的计算机事业打下基础;我期望他们从这套书学到的远不仅仅是作者传授的知识,还要学到他严谨的治学态度、孜孜以求的探索精神和创新意识,更期盼他们以此为基础去创新、去攀登知识的巅峰。

作者计划,这套书共有7卷,但迄今他真正完成的还只有这头3卷。不过他已在全力以赴赶写第4和第5两卷,希望在两三年内使它们问世。最后,再争取把第6卷和第7卷全部完成。我期盼的是能够看到这全套书的大功告成,而我自己也能够把它们全部译完,奉献给我国的读者,这也算是我对我国计算机事业所做的一点点贡献吧!

顺便提到,本书得以问世,和许多人的帮助是分不开的。在这里,谨向为本书付出了劳动的江门五邑大学的孟亚老师,我的研究生杜民、胡茂伟以及暨南大学计算机科学系96级和97级的张永源、吕双欢、卢国旺等同学表示感谢。

并以此书献给已故中国科学院学部委员、我国计算机事业的先驱者之一、为我国计算机事业的发展做出重大贡献的我的老师王湘浩教授。

## 前 言

*Here is your book, the one your thousands of letters have asked us to publish. It has taken us years to do, checking and rechecking countless recipes to bring you only the best, only the interesting, only the perfect. Now we can say, without a shadow of a doubt, that every single one of them, if you follow the directions to the letter, will work for you exactly as well as it did for us, even if you have never cooked before.*

这就是你们要的书,是你们数不清的来信要求我们出版的书,它已花了我们多年的时光,我们一遍又一遍地咀嚼每一个细节,检查每一道食谱,惟一的愿望是把最好的、最有趣的和最完美的菜肴奉献给你们。

现在我们可以毫无疑问地说,如果你精心仿效,即使你以前从未烹饪过,本书任何一道食谱都将如它为我们提供的那样,也为你提供最精致美味的食物。

—McCall's Cookbook (1963)

为数字计算机编写程序的过程是饶有趣味的,因为它不但具有经济和科学价值,也是犹如赋诗或作曲那样的美学实践。本书是多卷集的第1卷,旨在对读者进行作为熟练的程序员所必需的各种技巧的训练。

以下各章并不打算作为计算机程序的入门介绍。我们假定在此之前,读者已经有某些经验。先决条件实际上很简单,但初学者需要时间和实践才能理解数字计算机的概念。读者应具有:

a) 一台存储程序式数字计算机如何工作的概念;但不必是电子学方面的,而是指令如何被保存于机器存储器中以及逐个地被执行的方式。

b) 把问题的解法翻译成为计算机能“理解”的明确术语的能力。(这些机器没有普通感觉,它们只能按照告知它们的要求不折不扣地去做。当人们开始试图使用计算机时这是最难以掌握的概念。)

c) 最基本的计算机技术的某些知识,例如循环(重复地执行一组指令),使用子程序以及使用下标变量的某些知识。

d) 普通计算机术语——“存储器”、“寄存器”、“位”、“浮点”、“溢出”、“软件”等方面的少量知识。在正文中未予定义的大多数词,在每卷末尾的索引中都给出简明的定义。

这四个先决条件也许可以概括成一个要求,即读者至少应该为一台计算机编写并调试过至少四个程序。

我力图使这一丛书满足多方面的需要。首先,这些书是一套参考书,它汇总了在若干重要领域的知识。其次,它们可以作为计算机科学和信息科学领域的大学教材或自学读物。为了实现这两方面的目标,我在本书中加入了大量的习题并为大多数习题附上了答案。我还尽力用实例充实本书的内容,而避免作含糊空泛的评论。

这套丛书的对象,不是对计算机仅仅有一时兴趣的那些人们。但这绝不是说,这部书仅仅是为计算机的专业人员写的。事实上,我的主要目标之一是使这些程序设计技术可以被工作在其它领域的许多人掌握;这些人能够有效地使用计算机,却没有时间搜寻隐藏在各种技术杂志中的必需信息。

我们可以把这些书的主题称做“非数值分析”。计算机在传统上是同求解数值问题,例如一个方程的根,插值与积分等等的求解与计算相关联的。但在这里,我们并不处理这样一些课题,除非是顺便提及。数值计算机程序设计是极为有趣和迅速发展的领域,因而已有许多这方面的书出版。然而,自20世纪60年代初期开始,计算机已经更频繁地用于数值仅是偶尔出现的一些问题上;这里所使用的是计算机的判定能力,而不是它的算术运算能力。在非数值问题中,我们也用一些加法和减法,但我们很少需要乘法和除法。当然,即使是主要对数值计算机程序设计感兴趣的人也将非数值技术的学习中获益,因为非数值技术在数值程序中同样存在。

非数值分析的许多研究成果分散于许多技术杂志中。我的方法是通过研究那些最基本的技术,来提取大量文献的精华。所谓最基本是指它们可被应用于许多类型的程序设计场合。我尽力把这些思想上升为“理论”,并解释如何把这一理论应用于广泛的实际问题中。

当然,“非数值分析”是关于这个研究领域的极其负面\*的名称,最好应该有一个正面的描述性的术语来表征这个课题。“信息处理”对于这个内容而言太宽了,而“程序设计技术”又太窄了。因此,我希望以算法分析作为这套书所涵盖课题的恰当名称,这一名称隐含着“特定计算机算法性质的理论”的意思。

整套丛书,以“计算机程序设计艺术”为书名,其总目录如下:

第1卷 基本算法

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 组合检索

---

\* 因为用的是在“数值”之前加上“非”这样一种表述。——译者注



## 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 卷所处理的课题相当之大,它实际上包含 3 个分册(卷 4A, 4B 和 4C)。有关更专门课题的另外两卷也在准备中:第 6 卷语言理论(第 11 章)和第 7 卷编译程序(第 12 章)。

我从 1962 年开始按照这个章次来写一本书。但我很快发现,更重要的是要深入地讨论这些课题而不是浅尝辄止。我写出的文稿的篇幅使我意识到,每章所含的内容都已足以作为一学期的大学课程,因此把这部书分成几卷出版是明智的。我知道,在一本书中仅写一两章是很少见的,但我决定仍保留原来的章目以便于交叉参考。我也计划出一本第 1 卷到第 5 卷的缩写本,专门用作本科计算机课程的更通用的参考书或教材,其内容是这些书内容的一个子集,略去了其中更为专门的信息。在缩写本中的章目仍和全集中的一样。

本卷可看做整个丛书的一个“交叉点”,意思是,它包含了在所有其它书中用到的基本材料。另一方面,第 2 卷至第 5 卷可以彼此独立地阅读。第 1 卷不仅可作为同其它卷相关联的一本书,而且还可以作为数据结构(重点是第 2 章的内容)、离散数学(重点是 1.1, 1.2, 1.3.3 和 2.3.4 等节的内容)或者机器语言程序设计(重点是 1.3 和 1.4 节的内容)等课题的大学教材或自学教材。

在写这些章节时,我的宗旨不同于当代许多有关计算机程序设计的书,我并不试图教读者如何使用某些人的软件,我所关心的是教人们写更好的软件本身。

我原来的目标是把读者带到所讨论的每一课题的知识的前沿。然而,要在经济上可以盈利的一个领域保持不落伍是极其困难的,而且计算机科学的迅速发展使这一梦想已不可能实现。这一课题已经变成一个巨大的花毯,它由来自全世界的数以万计的智者所奉献的数以万计的精妙结果织成。因此我的新的目标已变成专注于那些“经典”的技术(它们可能在今后数十年间仍保持其重要性),并尽我之所能来描述它们。特别是,我已试图来跟踪每一课题的历史,并且为未来进展提供一个坚实基础;我尝试选择简明并同当前用法一致的术语;我试图把既优美而又易于表述的有关顺序计算机程序设计的所有已知的思想都包括进来。

关于本丛书的数学内容还有几句话要说一说。我已把这些材料组织成,使得只需有高中代数知识的人就可读懂它,他们只须跳过更深入的数学部分即可。同时,那些数学功底好的读者将能学习同离散数学有关的许多有趣的数学技巧。为了兼顾这两方面的需要,我对每个习题都指定级别,并把那些主要是数学的习题特地标了出来,同时,在大多数章节中,把主要的数学结果放在它们的证明之前来叙述。这些证明或者被留作习题(其答案可在一个独立的部分找到),或者在同一节的末尾中给出。

主要是对程序设计而不对相关的数学感兴趣的读者可以在一旦对数学内容感到困难时就停止阅读它的大部分内容。但另一方面,面向数学的读者将在这里发现丰富、

有趣的内容。有关计算机程序设计的许多发表过的数学知识都是有错误的。因此本书的目的之一是向读者讲授有关这一课题的适当的数学内容。由于我自认为是一名数学家,因此尽我所能来维护数学的完整性是我的义务。

对于本丛书中的大多数数学内容而言,有初等微积分的知识就足够了。因为需要的大多数其它理论都是由此而建立起来的。然而,有时我确实需要使用复变函数论、概率论、数论等等较深入的定理。在这种情况下,我会列出讨论这些课题的适当教材供读者参考。

在编写这些书时我要做出的最难的决策在于介绍这些不同技术的方式,流程图和算法的逐步非正规描述的优点是众所周知的。关于这方面的讨论,见论文“Computer Drawn Flowcharts”,ACM Communications,第6卷(1963年9月),555~563。同时,一种精确的语言对于描述任何计算机算法也是必要的,因此我需要决定,为此目的是使用一种代数语言例如ALGOL或FORTRAN呢,还是使用面向机器的语言。或许许多当今的计算机专家都不同意我使用一种面向机器的语言,然而我已深信,由于下列原因,它肯定是一个正确的选择:

a) 程序员在很大程度上受他写程序时所用的语言的影响;有一个压倒性的趋势,就是宁愿使用在语言中最简单的结构,而不会去使用对于机器而言最好的那些东西。通过理解一种面向机器的语言,程序员就将趋向于使用有效得多的方法;这更接近于现实。

b) 除去少数例外,我们所要求的程序都是比较短的,因此只要有一台适当的计算机,理解这些程序将毫无麻烦。

c) 对于讨论诸如共行程序(coroutine)链接、随机数生成、多精度算术以及涉及存储器的有效使用等许多问题的重要的低级细节而言,高级语言是不适当的。

d) 一个真正对计算机感兴趣的人,应当很好地学习机器语言,因为它是一台计算机的基本部分。

e) 无论如何,作为在许多例子中叙述的软件程序的输出,某种机器语言将是需要的。

f) 新的代数语言每5年就走红和过时,而我想强调的是永恒的概念。

我承认,从其它观点来看,以高级语言来写程序是稍微容易一些的,而且调试程序要更容易。确实,自1970年以来,对于我自己的程序,我也很少使用低级的机器语言,因为计算机是如此之快和如此之强了。然而对于本书中我们感兴趣的许多问题而言,程序员的技巧是至为重要的。例如,某些组合计算需要被重复1万亿次。因此如果我们能从它们的内循环挤出1微秒来,就可以节约出11.6天来。类似地,对于在许多计算机装置中每天要被用上许多次的软件编写来说,作出额外的努力是值得的,因为这个软件只须编写一次。

在做出了使用面向机器的语言的决策之后,该使用哪一种语言呢?我可以选择一个特定的机器X的语言,但如果那样,不具有机器X的人们就会认为本书是仅为使用X的人们写的。而且,机器X可能有大量的特性,它们同本书的内容全然无关,但我们

仍须加以说明。况且在两年内,机器  $X$  的厂家将推出机器  $X+1$  或  $10X$  来,那机器  $X$  将不再使任何人感兴趣了。

为避免陷入这种窘境,我试图设计一种具有非常简单的操作规则(比如说,只要花上一小时便可学会)的“理想”计算机,而它又非常类似于真实的机器。学生毫无理由害怕领会一种以上的计算机的特性;一旦掌握了一种机器语言,其它语言就很容易吸收了。确实,认真的程序员可以预期,在他们的事业中会遇到许多不同的机器语言。因此,一台虚构的机器剩下的惟一缺点是执行为它编写的任何程序的困难。幸而,这实际上不是一个问题。因为许多志愿者已经站出来为这假想的机器写了模拟程序。这样的模拟程序对于教学的目的而言是理想的,因为它们比起一台实际的计算机来更易于使用。

我试图在每个主题中引用早期的最好论文及最新的作品。当引用参考文献时,我都使用这些期刊的标准缩写,但以下最普遍被引用的杂志例外,它们的缩写如下\*:

*CACM* = Communications of the Association for Computing Machinery(《美国计算机协会通讯》)

*JACM* = Journal of the Association for Computing Machinery(《美国计算机协会杂志》)

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

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

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

*SICOMP* = SIAM Journal of 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* (《纯粹和应用数学杂志》)

例如,“*CACM* 6 (1963), 555 ~ 563”表示上页中提到的参考文献;我也使用 *CMath* 来代表 *Concrete Mathematics* 这本书,它在 1.2 的引言中被提到。

这一套书中的许多技术内容还出现于习题中。当一道较难的习题背后的思想不属于我自己的时候,我尽量提及提出该思想的人来。对于相应的参考文献的引用通常则在该节伴随的文字中给出,或者在习题的答案中给出,但在多数的情况下,习题都基于未公开发表过的材料,对于它们就无法进一步给出参考文献了。

在编写这些书的过程中,我获得了许多人的帮助,对此我深为感激。首先我要感谢我的妻子 Jill,感谢她无限的耐心,感谢她为本书绘图,感谢她数不尽的各种帮助。其


---

\* 鉴于参考文献直接引用原文更便于读者检索,本书保留参考文献原有文字而不译出。读者可从书末索引与词汇表中查到文献和人名的译名,或从所附网址查找有关信息。对于本书涉及的人名,除极少数众所周知者外,均直接使用原文而未译出。——本书责任编辑

次,我要感谢 Robert W. Floyd,他自 20 世纪 60 年代以来为扩充本书的内容奉献了大量的时间。还有上千人也提供了重要的帮助——光列出他们的名字就将是一本书了。他们中的许多人善意地允许我使用他们迄今为止未发表的作品。我在 Caltech(加州理工学院)和 Stanford(斯坦福)的研究有许多年都得到国家科学基金和海军研究办公室的慷慨支持。自 1962 年我开始本项目以来,Addison-Wesley 一直提供了极好的帮助与合作。我知道向每一个人表示感谢的最好方法是通过这一出版物展示出,他们的投入成就了这些书,我想,这些书正是他们要我写成的那个样子。

在花费了 10 年时间研制用于计算机排版的 T<sub>E</sub>X 和 METAFONT 系统之后,现在我已经有能力实现在开始这一研制工作时就有了的理想,即应用这些系统于《计算机程序设计艺术》中。最终,本书的全部文本都以电子文件的形式存入到我的个人计算机当中,以使它可以容易地适合于未来的印刷和显示技术。新的设置使我得以作字面上的数以千计的改进,它们是我早就想付诸实施的。

在这一新版本中,我已经字斟句酌地校阅每段文字,试图保留我原来句子的蓬勃朝气,而或许又增加上今日的深思熟虑。已经增加了几十道新习题,还有几十道老的习题也配上新的和改进了的答案。

然而,《计算机程序设计艺术》仍然在写作之中,因此本书的某些部分带有“正在施工”的图标,以表示对该部分内容尚未被更新这一事实的歉意。我的文件夹里已挤满了重要的材料,因而我打算把它们包括在最后的、光荣的第 1 卷的第 4 版中,或许在从现在开始的 15 年内吧。但我必须首先完成第 4 卷和第 5 卷,除非绝对必须,我不想把它们的出版时间拖延得更长。

准备这一新版本的大部分艰苦工作是由 Phyllis Winkler 和 Silvio Levy 完成的,他们熟练地敲入和编辑了第 2 版的文本,还有 Jeffrey Oldham,他把几乎所有原来的插图都转换成 METAPOST 的格式。我已经改正了机敏的读者们在第 2 版中发现的错误(以及某些,天啊,竟无人注意到的错误),我已试图避免在新的内容中引进新的错误。然而,我设想,还会有某些错误存在,我要尽可能彻底地改正它们。因此我将高兴地对每一个技术、排版或历史错误的头一个发现者支付 2.56 美元。版权页上的网页包含有已向我报告的所有错误的更正\*。

Stanford, California

1997 年 4 月

D. E. K

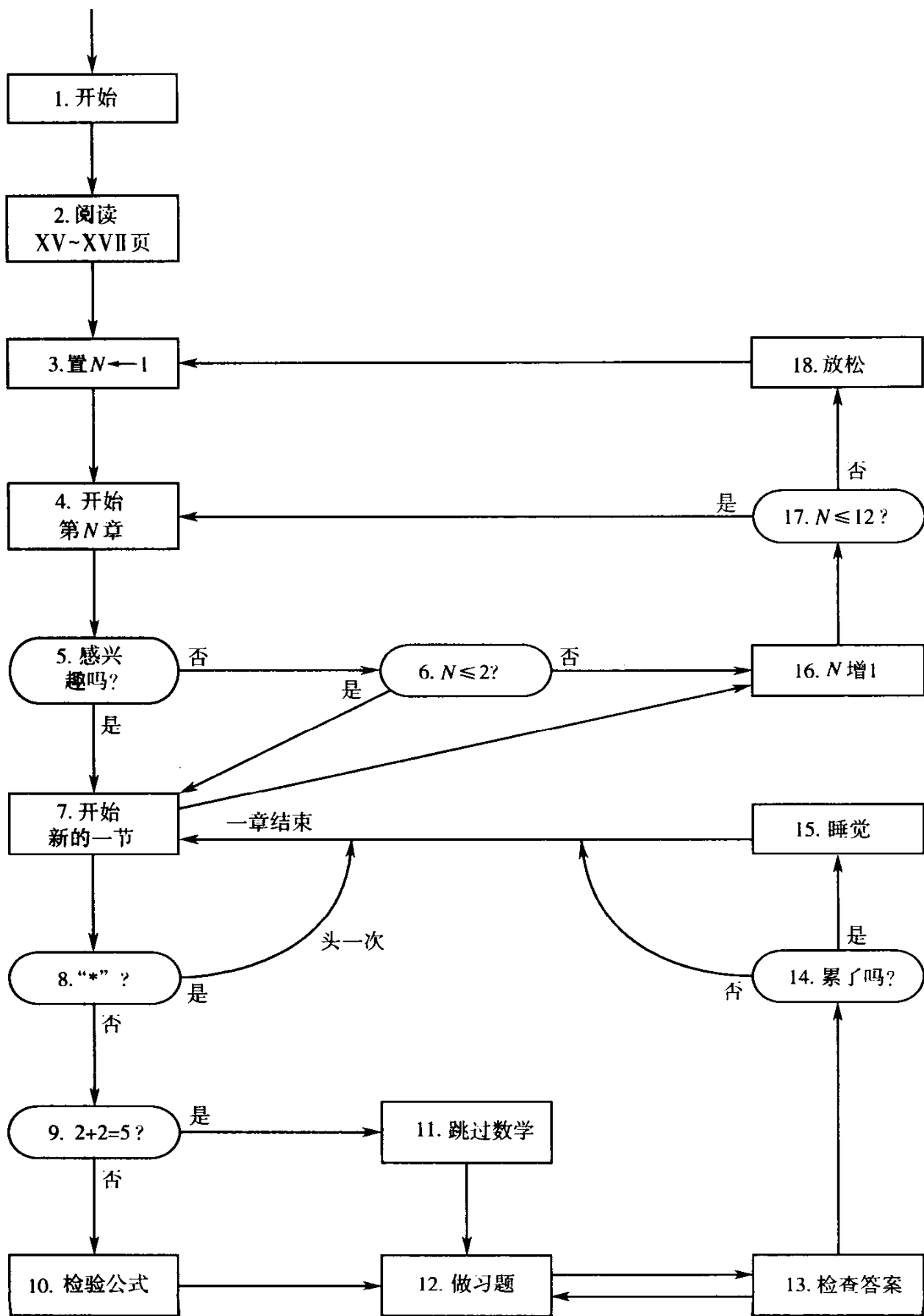
*Things have changed in the past two decades.*

一晃 20 年,形势变了。

—BILL GATES(1995)

---

\* 中文版已按 2001 年 10 月的最新勘误表更正了原书的错误。网页 <http://www.ndip.com.cn/computer/>上也将列出对中文版的勘误表,我将很诚挚地对发现中文版错误的读者表示感谢,但我不敢保证也支付您 2.56 美元。——本书责任编辑



阅读本丛书的流程图

## 阅读本套书的步骤

1. 开始,先阅读本步骤,除非你已经读过了。继续忠实地遵循这些步骤。(这一过程的一般形式及与之相应的流程图贯穿全书。)

2. 阅读后文关于习题的说明。

3. 置  $N$  等于 1。

4. 开始阅读第  $N$  章,不要读该章开头的引语。

5. 你对这章的主题感兴趣吗? 若感兴趣,转到第 7 步;若不感兴趣,转到第 6 步。

6.  $N \leq 2$  吗? 若否,转到第 16 步;若是,随便地浏览这一章。(第 1 章和第 2 章包含有重要的介绍性材料,并且也是基本的程序设计技术的综述,你应当至少浏览关于符号和关于 MIX 的部分。)

7. 开始阅读本章的下一节,但若你已读完本章,就转第 16 步。

8. 这一节带有“\*”吗? 若是,则在头一次阅读时,你可略去这一节(这里包含有较为专门的课题,这些课题是有趣的,但并非是必要的);转到第 7 步。

9. 你对数学有兴趣吗? 如果你对数学完全不懂,转到第 11 步;否则转到第 10 步。

10. 校验这一节中所做的数学推导(并把错误之处告知作者),转到第 12 步。

11. 如果这一节充满了数学计算,你最好略去这些推导不读。然而,你应当熟悉这一节的基本结果。这些基本结果通常在开始处附近叙述,或者在困难部分的末尾用 5°斜叙述。

12. 按照(你在第 2 步中读到的)关于习题的说明中给出的提示做这节中推荐的习题。

13. 在你已经自认为满意地把题目做完之后,将你的答案与本书后面相应的答案进行核对(如果对于这个问题有答案的话)。也读一读你没有时间做的习题的答案。注:在大多数情况下,在做第  $N+1$  题之前,先读第  $N$  题的答案是合理的,所以第 12 步和第 13 步通常是同时做的。

14. 累了吗? 若否,返回第 7 步。

15. 去睡觉,醒来后返回第 7 步。

16.  $N$  增 1,如果  $N = 3, 5, 7, 9, 11$  或 12,开始下一卷。

17. 若  $N$  小于或等于 12,返回第 4 步。

18. 祝贺你,现在试试来说服你的朋友购买第 1 卷,并开始阅读它,而且,返回第 3 步。

*Woe be to him that reads but one book.*

但愿他不只读一本书而已。

—GEORGE HERBERT, *Jacula Prudentum*, 144 (1640)

*Le défaut unique de tous les ouvrages*

*c'est d'être trop longs.*

它惟一的缺点就是太长了。

—VAUVENARGUES, *Réflexions* 628 (1746)

*Books are a triviality. Life alone is great*

书是平凡的；只有生命才是伟大的。

—THOMAS CARLYLE, *Journal* (1839)

## 关于习题的说明

本套书的习题既可用于自学,也可用于课堂练习。无论是谁,如果想纯粹地通过阅读,而不将所阅读的信息应用到特定问题上,并由此牵引思考先前阅读的内容,就想学到一门学问,纵然可能的,那也是很困难的。其次,对于我们自己的发现,我们总是领会得最透。因此,习题形成了这一套书的一个重要部分;我着意使这些习题含有丰富的信息,而且也尽量选择既有趣又有启发性的习题。

在许多书里,容易的习题被随机地混杂于极端困难的问题当中。有时这是不合适的,因为读者预先想要知道一道习题花多少时间——否则他们可能跳过这些习题了事。这一情况的典型例子是由 Richard Bellman 所著的 *Dynamic Programming* 一书。这是一本重要的先驱性的论著,其中在某些章的末尾,在“习题和研究题”的标题下,把一组问题堆集在一起,而且一些极其平凡的问题出现在一些深入的、还未被解决的问题当中。据说,有人曾问过 Bellman 博士,怎样区分习题和研究题。他回答说:“如果你能解决它,那它就是一道习题,否则它就是一个研究题。”

但在这一类书里把研究问题和很容易的习题都包括进来确有其道理;因此,为使读者免于陷入确定哪是习题哪是研究题的困境,我给习题注明了等级分,以指出困难的程度。这些分数大体具有下列意义:

- | 分数 | 解释   |
|----|--|
| 00 | 一个极其容易的问题,如果你已理解正文的内容,就可能立即做出回答。这样一道题几乎总是可以“眉头一皱”就把它做出。                  |
| 10 | 一个简单的问题。它要求你去思考刚刚学过的内容,但绝不意味着是困难的。你应当有能力在顶多一分钟之内就把它做出。在获得解答的过程中可能要用到笔和纸。 |
| 20 | 一个普通的问题。它检查你对正文内容的基本理解,但你可能需要十五或二十分钟才能完整地回答它。                            |
| 30 | 一个中等难度或中等复杂的问题。这个题目可能需要两个小时以上的工作才能令人满意地解决。或者甚至更长时间,如果电视机在开着的话。           |
| 40 | 确实是一个十分困难或冗长的问题。在学校里,它将适合于作为一个学期的课程设计。一个学生应当有能力在相当长的时间里来解决它,但这个解不会是平凡的。  |
| 50 | 就作者在编写本书时所知,这是一个还未令人满意地解决的问题,尽管已经有很多人做了尝试。如果你已经找到了这样一个问题的答案,你应当          |



把它写出来发表。其次,本书的作者将乐意尽快地听到关于这一解答的消息(当然,假定它是正确的)。

通过在这个“对数”尺上的内插,其它分数的意义也就清楚了。例如 17 分将表示比普通的题更为简单的一道习题。一个随后被某个读者解决了的 50 分的题在本书后来的版本中将以 45 的分数出现,而且会在互联网上的勘误表中刊出(参见版权页)。

分数除以 5 的余数表示所要求的详细工作量。因此分数是 24 的习题可能要比分数是 25 的题花费更长的时间来求解,但后者将要求更多的创造性。

作者已经认真地试图指定精确的分数,但是提出问题的人要想确切地知道,所提问题对求解的另一个人难度如何,肯定是困难的;而且每个人都会有较其他人更为适应的问题类型。只能希望这些分数较好地反映了习题的困难程度,希望读者把它们当做一般的导引,而不应作为绝对的指标。

本书是为具有不同程度的数学训练和素养的读者写的;因此有些习题是特意为有更多数学基础的读者提供的。如果一道题的出题动机或涉及的数学概念超越了主要兴趣仅仅是算法编程本身的读者应掌握的程度,则在分数前边加上  $M$ 。如果一道习题的答案必须涉及在本书中未予提供的微积分或其它高等数学知识,则对这道题标以“ $HM$ ”的字母。“ $HM$ ”标记并不必定意味着困难。

某些习题的前边标有三角符“ $\blacktriangleright$ ”;这说明是特别有启发性的问题,因而特别予以推荐。当然,并不期望读者/学生来求解所有的习题,所以标出了那些看起来最有价值的部分(这并不意味着贬低其它习题!)。每个读者至少应该尝试去解分数是 10 或者更低的所有问题;三角符可以帮助指出应该对具有较高分数的哪些问题予以优先考虑。

在答案部分中已经列出大多数习题的解答,请明智地使用它们。在你已真正地做出努力亲自解决问题之前,不要去翻答案,除非你确实没有时间来这一特定的习题。在获得了你自己的解答或者对该题做了郑重的尝试之后,你可能会感到答案是有启发和有幫助的。给出的解答通常十分简短,作者认为你已经认真地通过自己的方法做过求解尝试,因而略去了其细节。有时解答所提供的信息比所要求的为少,但通常都是更多的。十分可能,你有比这里给的更好的答案,或者在答案中你发现一个错误。在这样的情况下,作者将乐意知道详细的情况。在本书以后的版本中将在适当地方刊登出这些改进了的答案以及提供这些解的人的姓名。

当做一道习题时,一般地你可以使用前边习题的答案,除非明确地禁止这样做。在对习题打分时,已经考虑到这一点了,因此很可能第  $n + 1$  题的分数比第  $n$  题还要低,尽管它把第  $n$  题的结果作为一个特殊情况包括进来。

代码	含义	00	立即可解的
		10	简单的(一分钟)
$\blacktriangleright$	特别推荐的	20	一般水平(一刻钟)
$M$	面向数学的	30	难度适中
$HM$	要求“高等数学”的	40	学期设计
		50	研究题