TURING 图灵原版计算机科学系列

PEARSON

计算机程序设计艺术

卷4A：组合算法（一）

（英文版）

[美] Donald E. Knuth 著

The Art of Computer Programming

Volume 4A: Combinatorial Algorithms, Part 1

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4A：组合算法（一）

（英文版）

阜阳师范学院图书馆
藏书

[美] Donald E. Knuth 著

The Art of Computer Programming

Volume 4A: Combinatorial Algorithms, Part 1

人民邮电出版社
北京

图书在版编目（C I P）数据

计算机程序设计艺术，卷4A ：组合算法. 1 = The Art of Computer Programming, Volume 4 A:Combinatorial Algorithms, Part 1 ：英文 / (美) 高德纳著. -- 北京 ：人民邮电出版社，2012.2
（图灵原版计算机科学系列）
ISBN 978-7-115-27050-4

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(2011)第258503号

内 容 提 要

《计算机程序设计艺术》系列是图灵奖得主高德纳倾尽心血进行的一项巨大的写作计划，目前已经完成了前三卷。这套书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和算法理论，对计算机领域的发展有着极为深远的影响。本书是该系列的卷4A，主要介绍了组合算法，内容涉及布尔函数、按位操作技巧、元组和排列、组合和分区以及所有的树等。

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4A：组合算法（一）（英文版）

◆ 著　　[美] Donald E. Knuth
责任编辑　王军花

◆ 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
北京铭成印刷有限公司印刷

◆ 开本：700×1000　1/16
印张：56.25
字数：900 千字　　2012 年 2 月第 1 版
印数：1 – 3 000 册　　2012 年 2 月北京第 1 次印刷

著作权合同登记　图字：01-2011-6038 号

ISBN 978-7-115-27050-4

定价：129.00 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

Original edition, entitled *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*, 978-0-201-03804-0 by Donald E. Knuth, published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2011 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 © 2012.

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。

前　　言

欲把一切好的内容都装进一本书中，显然是不可能的，哪怕只是想比较全面地涉猎主题的某些方面，多半也会导致篇幅急剧增长。

——Gerald B. Folland，“编者之角”（2005）

卷 4 的书名是组合算法，而当我拟书名时，曾特别想给它加一个副标题：我最喜爱的程序设计类型。但是，编辑们决定淡化个人的感情色彩，因此没有这么做。不过事实上具有组合风格的程序始终是我所偏爱的。

另一方面，我经常惊奇地发现，“组合”一词在许多人的头脑中意味着计算有难度。其实，Samuel Johnson 在他著名的英语词典（1755）中已经说明，对应组合这个名词“现在普遍用法不当”。同事们对我讲述种种不利事件时，总会说“事态的组合使我们无功而返”。对我而言，组合唤起的是纯粹喜悦之情，它却给其他许多人带来一片惊恐，原因何在?

的确，组合问题经常同非常巨大的数字相联系。Johnson 的英语词典中还引用了 Ephraim Chambers 的一段话，说用 24 个字母的字母表构成长度小于或等于 24 的词，总数会高达：

$$1\,391\,724\,288\,887\,252\,999\,425\,128\,493\,402\,200$$

用 10 个字母的字母表，相应的数字为 11 111 111 110，而当字母个数为 5 时，这个数字仅有 3905。所以，如果问题的规模从 5 增长到 10，再增长到 24 甚至更大，就必然出现“组合爆炸”。

在我的一生中，计算机一直在以惊人的速度变成越来越强大的工具。等到我写这些字句时，我知道所用的笔记本电脑的运算速度比我着手编写这套书时所用的 IBM Type 650 计算机快 10 万倍以上，而且现在这台电脑的存储容量也比那时大 10 万倍以上。明天的计算机还会更快并具备更大的容量。然而，这些惊人的进展没有减少人们对于回答组合问题的渴求，情况恰好相反。从前无法想象的如此快速的计算能力提高了我们的期待，并且激起我们更大的欲望——事实上，n 只要增加 1，组

试读结束：需要全本请在线购买： www.ertongbook.com

合问题的规模都有可能增加10万倍以上。

可以把组合算法非正式地定义为对排列或图这样一些组合对象的高速处理方法。我们向来试图找出一些模式或排列作为满足约束条件的最佳方法。这种问题的数量非常大，而且编写这类程序的技巧特别重要，很吸引人，因为有时只要一个好主意就可能节省几年乃至几百年计算机时间。

组合问题的优良算法具有巨大回报，这个事实激励了技术水平的突飞猛进。许多过去认为很难处理的问题现在可能迎刃而解，许多过去以优良著称的算法现在变得更好。大约从1970年起，计算机科学家们经历了所谓的“Floyd引理”现象：看似需用n^3次运算的问题实际上可能用$O(n^2)$次运算就能求解，看似需用n^2次运算的问题实际上可能用$O(n \lg n)$次运算就能处理，而且$n \lg n$通常还可以减少到$O(n)$。一些更难的问题的运行时间也从$O(2^n)$减少到$O(1.5^n)$，再减少到$O(1.3^n)$，等等。一般说来，剩下的问题依然是很困难的，但是已经发现它们有一些非常简单的重要特例。我曾经以为在我有生之年不会看到答案的许多组合问题，如今已经获得解决，而且那些突破的取得主要归功于算法的改进而不是处理器速度的提高。

截至1975年，这方面的研究进展神速，主要的计算机期刊上发表的大量论文竟然都是有关组合算法的。同时，这些进展不仅是由核心的计算机科学领域的研究人员取得的，而且大量成果也来自电子工程、人工智能、运筹学、数学、物理学、统计学等各领域的研究人员。我曾想尽快完成《计算机程序设计艺术，卷4》的写作，但是感觉自己就像坐在烧开了直冒汽的一壶水旁边，只不过我面对的是层出不穷的新主意的大爆炸！

这套书诞生于1962年之初，当时我天真地拟好一共12章的提纲。未经深思熟虑，我便决定用其中一章来简述组合算法，心想：“请看，大多数人利用计算机处理数字，而我还能够编写处理计算模式的程序！”在那个时期，对于已知的每个组合算法，都能很容易给出一个非常完备的描述。即使到1966年，当我把这本已经过度膨胀的书写好约3000页初稿的时候，其中第7章的内容还不到100页。我绝对不会想到，当初预计作为“沙拉”的小菜最终竟然会升格成一道主菜。

1975年兴起的组合学热潮，在越来越多人的参与下一浪高过一浪。新思想不断改善着旧思想，但是很少取而代之，或者使其过时。所以，我自然就得抛弃曾经的希望，已经不可能围绕这个领域撰写一本一劳永逸的书，把一切题材组织得井然有序，让需要解决组合问题的每个人“一册在手，别无他求”。各种各样可用的技术如雨后春笋般破土而出，我几乎对任何支节问题都不可能做到说：“这是最后的解决方案：故事结束了。”相反，我必须把自己严格限制在只阐明一些最重要的原理，而这些原理应该是迄今我所遇见的所有有效的组合方法的基础。现在我为卷4积累的原始资料是卷1至卷3全部资料的两倍还不止。

这些堆积如山的资料意味着实际上必须把我以往计划的“卷 4”变成若干卷。读者现在见到的是卷 4A。假设我身体健康，以后还会有卷 4B 和卷 4C，并且（天知道？）可能还有卷 4D、卷 4E……当然肯定不会出现卷 4Z。

现在的计划是通过自 1962 年以来积累的文档，尽我所能，系统地讲述（其实仍然有待进一步讨论的）组合方法。我无意追求完美，但是一定要把应有的荣誉归于所有那些提出关键思想的先驱，所以对于历史情况我不会惜墨如金。除此之外，凡是我以为在今后 50 年仍然具有重要性的某些内容，以及能用一两段文字简洁说明的某些内容，我都不能割爱。反过来，我没有把那些需要长篇累牍证明的艰深话题收录进来，除非它确实是基础性的。

不错，组合算法这个领域显然是广阔的，我不能顾及它的所有方面。那么，我忽视了什么最重要的东西呢？我以为我的最大盲点是在几何上，因为我最擅长的始终是呈现和操作代数公式而非空间对象。所以，在这几本书中，我不准备讨论与计算几何有关的组合问题，如球的密堆积，n 维欧几里得空间中数据点的分类，甚至也不讨论平面内的 Steiner 树问题。更重要的是，我力求避开多面体组合学，以及主要基于线性规划、整数规划或半定规划的各种方法。那些内容在有关这个主题的其他很多书中已被较好地涉及，而且它们依赖于几何直觉知识。单纯从组合问题展开讨论对我来说是更容易理解的。

我还必须承认，对于那些仅在渐近意义下有效的组合算法，以及优越性能直到问题规模超越乎想象时才开始显现的组合算法，我不太感冒。现在有大量出版物讨论这类算法。我能理解有的人喜欢思考极限问题，认为它是个智力挑战并且能带来学术声望，但是我这本书对于我自己在实际程序中不考虑使用的任何方法，一般只是轻描淡写。这条规则的运用自然也有例外，特别是对那些处于主题中心的基本概念就不是这样。（某些不实用的方法实在是非常优美或者含义深刻，让我不能割舍，还有些方法则被我作为反例引用。）

此外，在这套书的前几卷中，我有意讲的都是顺序算法，尽管计算机的并行计算能力日益增强。对于哪些并行算法可能在今后 5 年或者 10 年中会很有用，我判断不好，更不用说 50 年后了，所以我乐于把这样的问题留给那些比我聪明的人。明日才华横溢的程序员们应该具备什么样的顺序算法知识，单是这个问题就足够考验我自己的能力了。

在陈述这些材料时，我需要作出的一个主要决定是按问题还是按方法组织它们。例如，卷 3 中的第 5 章专门讨论一个问题，即数据排序，我们从不同方面应用了 20 余种方法。相比之下，组合算法涉及许多不同的问题，而解决问题的方法则少很多。最终我断定，采用一种混合策略会比任何一种单纯的方法能够更好地组织材料。于是在这几本书中，7.3 节处理求最短路径问题，7.4.1 节处理连通性问题，但是其他许

多节则专门讨论基本方法，如布尔代数的应用（7.1 节）、回溯（7.2 节）、拟阵理论（7.6 节）或者动态规划（7.7 节）。著名的流动推销员问题，以及与覆盖、着色和填充有关的其他经典组合问题，没有单辟章节讨论，但是它们在书中多次出现，每次都用不同的方法处理。

我已经提到过组合计算技术的巨大进步，但是我并不是暗示人们已经解决了所有的组合问题。在计算机程序的运行时间难以掌控之际，程序员们不能指望从本书中找到无坚不摧的“银弹”。这里描述的方法通常会比一个程序员先期尝试的方法快很多，但是让我们面对这样一个现实：组合问题一不小心就会迅速地成为巨大的问题。我们甚至可以严格证明，就连一个自然的小问题也不存在现实的可解，尽管原则上它是可行解的（参见 7.1.2 节的 Stockmeyer 和 Meyer 定理）。还有些情况下，我们尚不能证明一个给定问题不存在合适的算法，只是知道可能没有这样的算法，因为任何有效的算法将会产生一种有效方法，帮我们解决一大批难倒世上无数英雄好汉的问题（参见 7.9 节关于 NP 问题的讨论）。

经验表明，新的组合算法将不断涌现，以解决新出现的组合问题以及老问题的变型或特例，并且人们对于这样算法的期望也会继续提高。当程序员总是面临这样一些挑战时，计算机程序设计艺术必将会不断达到新的高度。不过，今天的方法也可能仍然会长期发挥重大作用。

本书的大部分内容是相对独立的，但与卷 1 至卷 3 的内容时有关联。前几卷深入地讨论了机器语言程序设计的底层细节，所以本书中的算法常常是在抽象级别上说明，与任何具体机器无关。然而，组合程序设计确有某些方面与过去未出现过的底层细节密切相关。出现这种情况时，书中相应例子都是基于 MMIX 计算机的，这种计算机替代了卷 1 前几版中定义的 MIX 计算机。关于 MMIX 的详细材料在卷 1 的一本补充读物中介绍，即 *The Art of Computer Programming, Volume 1, Fascicle 1*，其中包含 1.3.1 节、1.3.2 节，等等。另外，这些材料也可以从因特网上获得，配套的汇编程序和模拟程序也可从网上下载。

另外一种可以从网上下载的资源是称为《斯坦福图库》（*Stanford GraphBase*）的一组程序和数据，它在本书的例子里经常引用。我鼓励读者多利用它，以便既高效又快乐地学习组合算法。

顺便说一句，在我撰写第 7 章开头这个前言的时候，我很高兴地说明，书中提到了我的博士论文导师 Marshall Hall, Jr.（1910—1990）的一些成果，以及 Hall 的论文导师 Oystein Ore（1899—1968）的一些成果，还有 Ore 的论文导师 Thoralf Skolem（1887—1963）的一些成果。至于 Skolem 的论文导师 Axel Thue（1863—1922）的一些成果，我在第 6 章中已经作过介绍。

有几百位读者帮助我查出在这卷书几份初稿中遗留的大量错误，谨对他们表示

特别的感谢，这些稿子原先公布在因特网上，后来又印成几册平装书。特别是 Thorsten Dahlheimer、Marc van Leeuwen 和 Udo Wermuth 三人的大量意见对本书产生了很大影响。但是，我担心还有其他错误隐藏在汇集成卷的字里行间，希望能尽快改正它们。因此，我乐于对首先发现任何一处技术错误、排版错误或者史料错误的人奖励 2.56 美元。在 http://www-cs-faculty.stanford.edu/~Knuth/taocp.htm/中包括一张勘误表，列出了当前已知的所有错误的修正。

D. E. Knuth

2010 年 10 月于加州斯坦福

在第 1 版序言中，我曾请求读者不要专门注意辞典中的错误，现在我反倒希望自己未这样说过，而且要感谢对我的请求置之不理的那些读者。

——Stuart Sutherland,《国际心理学辞典》(1996)

我自然应该对遗留下的错误负责——不过在我看来，我的朋友们本来还可以发现出更多错误。

——Christos H. Papadimitriou,《计算复杂性》(1994)

我愿意从事种种不同领域的工作，以便使我的错误分散得更稀疏。

——Victor Klee (1999)

关于参考文献的注释　若干经常引用的杂志和会议会刊有特别的代码名称，它们出现在书后的 Index 和 Glossary 中。但是各种 IEEE 会刊的引用中包含一个代表会刊类别的字母代码，置于卷号前，用粗体表示。例如“*IEEE Tranc.* **C-35**”是指 *IEEE Transactions on Computers*, Volume 35 (《IEEE 计算机会刊》, 第 35 卷)。IEEE 现在不再使用原来这些简便的字母代码，其实这些代码不难辨认：“**EC**”曾经代表“电子计算机”，“**IT**”代表“信息论”，“**SE**”代表“软件工程”，“**SP**”代表“信号处理”等，“**CAD**”的含义是“电路和系统的计算机辅助设计”。

类似“习题 7.10-00”这样的写法是指 7.10 节中一道还不知道题号的习题。

关于记号的注释　对于数学概念的代数表示，简单而直观的约定始终会促进发展，尤其是当世界上多数研究人员使用一种共同符号语言的时候。可惜在这方面，组合数学当前的事态多少有些混乱，因为同样一些符号在不同的人群中有时完全代表了不同意义；在比较狭窄的分支领域从事研究工作的某些专家，也会在无意中引进了彼此冲突的符号表示。计算机科学——它与数学中许多主题相关——应尽可能采用内部一致的记号来避开这种危险。所以我经常不得不在若干对立的方案中作出

选择，明知结果不能使人人满意。我尽全力选用自以为是未来最好的记号，靠着多年的经验以及同事之间进行的讨论，还经常在不同方案之间反复试验，找出适用的记号。在其他人尚未认同对立的方案时，通常有可能找到适用的共同约定。

附录 B 给出了本书中使用的所有主要记号的综合索引，其中不可避免也包含若干还不是标准的记号。如果读者偶然遇见一个有点奇怪或不好理解的公式，基本上可通过附录 B 找到说明我的意图的章节。不过我仍然应该在此举出几个例子，供你初次阅读本书时加以留意。

- 十六进制常数前面冠一个#字符号。例如，#123是指$(123)_{16}$。
- “非亏减”运算$x \dot{-} y$有时称为点减或者饱和减，结果为$\max(0, x-y)$。
- 三个数$\{x, y, z\}$的中位数用$\langle xyz \rangle$表示。
- 像$\{x\}$这样含单个元素的集合，在文中通常简单地用x表示，例如$X \cup x$或$X \backslash x$。
- 如果n是一个非负整数，n的二进制表示中取1的位数记为νn。另外，如果$n>0$，n最左边的1和最右边的1分别用$2^{\lambda n}$和$2^{\rho n}$表示。例如，$\nu 10=2$, $\lambda 10=3$, $\rho 10=1$。
- 图G和图H的笛卡儿积用$G \square H$表示，例如，$C_m \square C_n$表示一个$m \times n$环面，因为C_n表示n个顶点的一个环。

NOTES ON THE EXERCISES

THE EXERCISES in this set of books have been designed for self-study as well as for 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. A motley mixture is, however, often 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," unless you're multitasking.

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. Maybe even twenty-five.

试读结束：需要全本请在线购买： www.ertongbook.com

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 (see page iv).

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 distinction 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.

Several sections have more than 100 exercises. How can you find your way among so many? In general the sequence of exercises tends to follow the sequence of ideas in the main text. Adjacent exercises build on each other, as in the pioneering problem books of Pólya and Szegő. The final exercises of a section often involve the section as a whole, or introduce supplementary topics.

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 printings 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
M	Mathematically oriented	*40*	Term project
HM	Requiring "higher math"	*50*	Research problem

EXERCISES

▶ **1.** [*00*] What does the rating "*M15*" mean?

2. [*10*] Of what value can the exercises in a textbook be to the reader?

3. [*HM45*] Prove that every simply connected, closed 3-dimensional manifold is topologically equivalent to a 3-dimensional sphere.

Art derives a considerable part of its beneficial exercise
from flying in the face of presumptions.
— HENRY JAMES, "The Art of Fiction" (1884)

I am grateful to all my friends,
and record here and now my most especial appreciation
to those friends who, after a decent interval,
stopped asking me, "How's the book coming?"

— PETER J. GOMES, *The Good Book* (1996)

I at last deliver to the world a Work which I have long promised,
and of which, I am afraid, too high expectations have been raised.
The delay of its publication must be imputed, in a considerable degree,
to the extraordinary zeal which has been shown by distinguished persons
in all quarters to supply me with additional information.

— JAMES BOSWELL, *The Life of Samuel Johnson, LL.D.* (1791)

The author is especially grateful to the Addison–Wesley Publishing Company
for its patience in waiting a full decade for this manuscript
from the date the contract was signed.

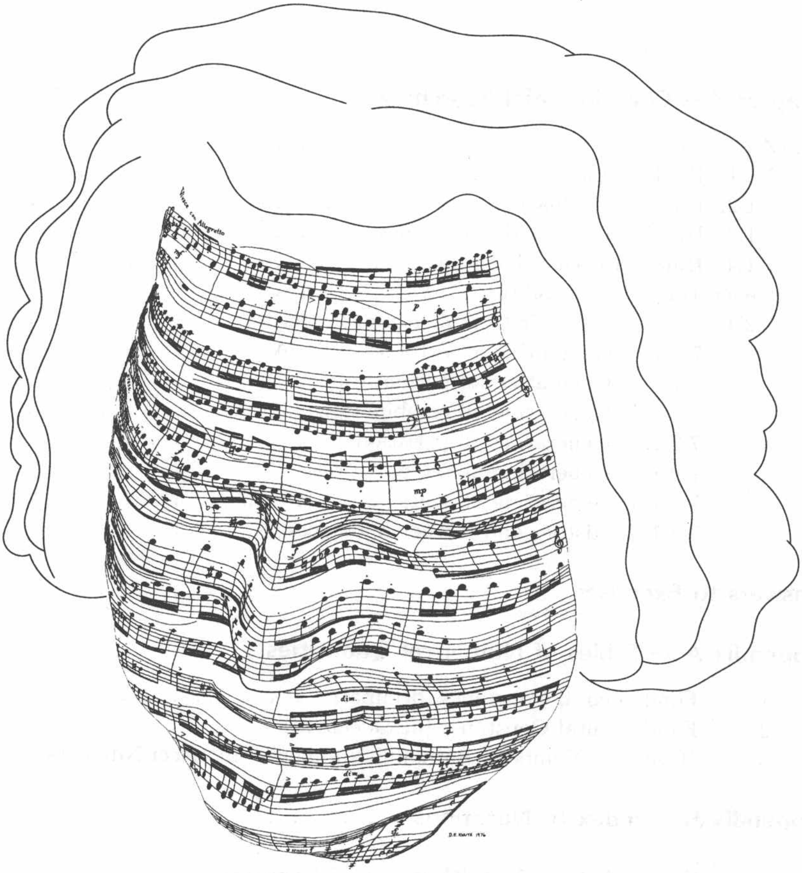
— FRANK HARARY, *Graph Theory* (1969)

The average boy who abhors square root or algebra
finds delight in working puzzles which involve similar
principles, and may be led into a course of study
which would develop the mathematical and inventive bumps
in a way to astonish the family phrenologist.

— SAM LOYD, *The World of Puzzledom* (1896)

Bitte ein Bit!

— Slogan of Bitburger Brauerei (1951)



Hommage à Bach.

CONTENTS

CHAPTER SEVEN

COMBINATORIAL SEARCHING

You shall seeke all day ere you finde them,
& when you have them, they are not worth the search.
— BASSANIO, in *The Merchant of Venice* (Act I, Scene 1, Line 117)

Amid the action and reaction of so dense a swarm of humanity,
every possible combination of events may be expected to take place,
and many a little problem will be presented which may be striking and bizarre.
— SHERLOCK HOLMES, in *The Adventure of the Blue Carbuncle* (1892)

The field of combinatorial algorithms is too vast to cover
in a single paper or even in a single book.
— ROBERT E. TARJAN (1976)

While jostling against all manner of people
it has been impressed upon my mind that the successful ones
are those who have a natural faculty for solving puzzles.
Life is full of puzzles, and we are called upon
to solve such as fate throws our way.
— SAM LOYD, JR. (1926)

COMBINATORICS is the study of the ways in which discrete objects can be arranged into various kinds of patterns. For example, the objects might be $2n$ numbers $\{1,1,2,2,\ldots,n,n\}$, and we might want to place them in a row so that exactly k numbers occur between the two appearances of each digit k. When $n=3$ there is essentially only one way to arrange such "Langford pairs," namely 231213 (and its left-right reversal); similarly, there's also a unique solution when $n=4$. Many other types of combinatorial patterns are discussed below.

Five basic types of questions typically arise when combinatorial problems are studied, some more difficult than others.

- i) Existence: Are there any arrangements X that conform to the pattern?
- ii) Construction: If so, can such an X be found quickly?
- iii) Enumeration: How many different arrangements X exist?
- iv) Generation: Can all arrangements X_1, X_2, … be visited systematically?
- v) Optimization: What arrangements maximize or minimize $f(X)$, given an objective function f?

Each of these questions turns out to be interesting with respect to Langford pairs.

试读结束：需要全本请在线购买： www.ertongbook.com