

TURING

图灵程序  
设计丛书

SE  
SHOEISHA

# 程序员的 算法趣题

[日] 增井敏克 / 著 绝云 / 译



Interesting Algorithm Puzzles  
for Programmers



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

TURING

图灵程序  
设计丛书

# 程序员的 算法趣题

[日] 增井敏克 / 著 绝云 / 译



*Interesting Algorithm Puzzles  
for Programmers*

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

程序员的算法趣题 / (日) 增井敏克著; 绝云译

. -- 北京: 人民邮电出版社, 2017.7

(图灵程序设计丛书)

ISBN 978-7-115-45923-7

I. ①程… II. ①增… ②绝… III. ①程序设计

IV. ①TP311.1

中国版本图书馆CIP数据核字(2017)第130676号

## 内 容 提 要

本书是一本解谜式的趣味算法书,从实际应用出发,通过趣味谜题的解谜过程,引导读者在愉悦中提升思维能力、掌握算法精髓。此外,本书作者在谜题解答上,通过算法的关键原理讲解,从思维细节入手,发掘启发性算法新解,并辅以Ruby、JavaScript等不同语言编写的源代码示例,使读者在算法思维与编程实践的分合之间,切实提高编程能力。

本书适合已经学习过排序、搜索等知名算法,并想要学习更多有趣算法以提升编程技巧、拓展程序设计思路的程序员,以及对挑战算法问题感兴趣、爱好解谜的程序员阅读。

---

◆ 著 [日] 增井敏克

译 绝 云

责任编辑 杜晓静

执行编辑 高宇涵 侯秀娟

责任印制 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>

北京鑫丰华彩印有限公司印刷

◆ 开本: 880×1230 1/32

印张: 9.75

字数: 320千字

2017年7月第1版

印数: 1-4000册

2017年7月北京第1次印刷

著作权合同登记号 图字: 01-2016-4462号

---

定价: 55.00元

读者服务热线: (010)51095186 转 600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字20170147号

站在巨人的肩上

Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上  
**Standing on Shoulders of Giants**



iTuring.cn

## 版 权 声 明

プログラマ脳を鍛える数学パズル

( ProgrammerNou wo Kitaeru Sugaku Puzzle : 4245-6 )

Copyright © 2015 by Toshikatsu Masui.

Original Japanese edition published by SHOEISHA Co., Ltd.

Simplified Chinese Character translation rights arranged  
with SHOEISHA Co., Ltd. through CREEK & RIVER Co., Ltd.  
and CREEK & RIVER SHANGHAI Co., Ltd.

Simplified Chinese Character translation copyright © 2017 by  
Posts & Telecom Press.

本书中文简体字版由 SHOEISHA Co., 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。

版权所有，侵权必究。

## 译者序

作为程序员，大家也许会有这样的“小洁癖”：特别不能忍受重复劳动，特别讨厌“人肉运维”。因此，只要做某件事需要花 90 秒以上的时间，那么就一定要通过写程序来完成这件事，哪怕写程序要花费半个小时。乍一听，这似乎是在浪费时间，然而这正是大部分优秀程序员的特质。一方面，如果是做重复的事情，计算机通常做得比人更快，准确率也更高；另一方面，写成程序之后，这些重复的流程更易于变更、管理和复用。事实上，正因为无数“有洁癖”的前辈们的伟大工作，才有了编译器，才有了百花齐放的编程语言，才有了欣欣向荣的 IT 产业。

不过，如果只是养成了“一言不合就写脚本”的习惯，与真正优秀的程序员仍然有很大的差距。同样是排序，不同的数据规模、不同的算法实现，性能表现都相差巨大。同样地，做同一件事，不同的程序员的解法和效率也天差地别。程序员圈内一直流传这样的说法：“优秀程序员的生产力可以达到普通程序员的十倍甚至成百上千倍。”ACM 圈子的高手，各种复杂精巧的算法信手拈来，应对极其复杂的问题时编码也如庖丁解牛，行云流水般顺畅；顶级的程序员甚至能创造世界级的工具，或者开创一种流派，影响大部分程序员的工作和思维方式。这种差距，真就像不同算法之间复杂度的差距一样明显，让人望而生畏。

见贤思齐。要怎么做才能步入“优秀程序员”的行列呢？抛开数学、各种计算机理论的基础不谈，也许最能量化程序员能力的就是“代码量”了。读更多优秀的代码，就能知道更多好的架构、好的算法；写更多的代码，解决问题的速度就更快，生产力也就更高。提高代码量这个简单粗暴的方法，效果的确立竿见影，于是乎一大批在线编程解题网站应运而生。而本书正是源于日本一个 IT 服务网站 CodeIQ 上的在线编程解题栏目“本周算法”。这个栏目的主编就是本书作者，他“寓教于题”，通过精心设计的问题向大家传授了很多算法、程序优化技巧甚至工程架构方面的经验等。

本书可以看作是一本算法书，与其他编程类、算法类图书最大的不同有两点：其一是所有问题都贴近生活和实际应用，兼具实用性和趣味性；其二是以虚拟的人物形象和实际的代码进行讲解，重点向读者演示不同思路、不同解决方案之间的区别和差距。公交车上如果设置自动找

零的装置，应该怎么实现？怎样实现一个简单的扫地机器人，让它尽量不要重复清扫某一个角落？如何串联和并联组合一堆电阻，使得最终电阻值逼近黄金分割值？像这样接地气、有意思的问题，书中比比皆是。讲解求斐波那契数列某一项的问题时，作者先由递归切入，后讲查表法优化，最后引出实际实现时需要处理数值溢出的问题。全书的讲解都像这样层层深入、条分缕析。

本书共 4 章，每一章都由很多问题构成。第 1 章讲的是最基础的二进制，通过实例帮助大家理解二进制，进而用二进制解决实际问题；第 2 章~第 4 章则分别从工程、算法和架构几个方面切入不同的算法优化案例。此外，个别问题下还会设置专栏，穿插一些作者在软件工程甚至人才培养等方面的理解和经验。

关于本书，最推荐的阅读方式是读完题先停下来想想解法。此时最好能打开电脑，打开编辑器，先试着把题做出来。做完之后再往下读，顺着作者的分析和解答细细体会问题背后的算法思路。书中每一个问题都汇集了作者以及 CodeIQ 网站大量用户的集体智慧，相信做完题再作对比，一定可以收获不少新的体会。如果您发现了更好的解法，希望您可以到图灵社区（<http://www.ituring.com.cn/>）或者本书的代码仓库（<https://github.com/leungwensen/70-math-quizzes-for-programmers>）上和大家分享交流，共同进步。

最后，成书不易，非常感谢图灵各位编辑的帮助和指导，也感念这将近一年时间里家人的理解和包容。

绝云

2017 年 4 月 5 日于杭州

# 前言

计算机的世界每天都在发生着深刻的变化。新操作系统的发布、CPU 性能的提升、智能手机和平板电脑的流行、存储介质的变化、云的普及……这样的变化数不胜数。

在这样日新月异的时代中，“算法”是不变的重要基石。要编写高效率的程序，就需要优化算法。无论开发工具如何进化，熟识并能灵活运用算法仍然是对程序员的基本要求。

程序员的工作说白了就是把需求变为程序。人们希望计算机做的事情就是“需求”，实现需求的就是“程序”。能满足需求的程序肯定不止一种，我们需要从中挑选出最优的程序。

这里的难点在于，怎样判断一个程序是不是最优的。不同的人对“优秀的算法”有着不同的理解。我认为，优秀的算法需要满足以下 3 点。

## (1) 高速

即使是简单实现后在处理上会花很长时间的程序，有时候转换一下角度进行优化，就能得到一个高速的版本。根据算法内容的不同，有时候优化的效果不仅仅是速度提升 2 倍、3 倍，甚至提升 100 倍、1000 倍的情况也不少见。

## (2) 简化

如何简化输入条件将会决定最终代码的复杂度。越是简单的程序，可维护性越高。

## (3) 通用

如果我们在实现程序时有意识地把通用的处理封装起来，那么就能把源代码用于其他问题或者工作需求上。如果实现的程序即便输入值或者参数发生变更，代码改动也很小，那么测试往往也能简化。

我很喜欢这么一句话：阅读量决定了学习能力的上限，写作量决定了学习能力的下限。这是因提出“百格计算”<sup>①</sup>而闻名的岸本裕史先生说的话，个人觉得这对编程也是适用的。要想磨练编程技巧只有两个途径：一是阅读代码，二是编写代码。

① 岸本裕史先生于昭和 40 年代（1965—1974 年）提出的儿童数学运算训练方法。在  $10 \times 10$  的格子的最左一列和最上一行随机填入 0~9 的数字，并在左上角的空格里指定运算符号（加减乘除），按照运算符号计算行与列中两个数字的运算结果，然后将结果填入该行与该列对应交叉的那个空格里。——译者注

不存在没有读过其他人写的代码的程序员。很显然，也不存在没有写过代码的程序员。越是编程技巧高超的开发者，读过的代码越多，写过的代码也越多。

数据结构和算法的学习尤为重要。多了解堪称无数先驱智慧结晶的算法，多亲身体会这些算法的效果对程序员非常重要。

本书为那些已经学习过排序、搜索等知名算法，并想要学习更多有趣的算法，进一步提升编程技巧的工程师准备了 69 道数学谜题形式的问题。

当然，本书中的算法并不是最优的。请怀着这样的心态，努力去思考更加优秀的算法。

## 致谢

在提供 IT 工程师业务技能评估服务的平台 CodeIQ (<https://codeiq.jp>) 上有一个名为“本周算法”的栏目。我在这个栏目中担任出题人，这一年多来每周都会公布一个算法趣题。本书中的问题就是出自这个栏目。当然，对于原问题，这里进行了些许修改和补充。感谢策划了该栏目的大成弘子女士、每周在我出题后进行检查的峯亚由美女士，以及其他 CodeIQ 相关的工作人员。

最后，还要感谢积极参与“本周算法”挑战的各位答题者。正是因为有你们，我才能持续不断地出题，最终让本书得以付梓。真的非常感谢大家。

增井敏克  
2015 年 10 月

## 本书概要

本书为 69 道数学谜题编写了解题程序。每个问题大致分为“问题页”和“讲解页”两部分，“问题页”从单页起。请各位先通读问题描述，并动手编写程序尝试解题。在这个过程中，具体的实现方法是其次，更重要的是思考“通过哪些步骤来实现才能够解决问题”。

翻过问题页就能看到思路讲解和源代码示例了。请留意自己编程时在处理速度、可读性等方面进行的优化，和本书上的源代码示例有什么不同。如果事先看了思路讲解和答案，就会失去解题的乐趣，所以这里建议大家先编程解题，再看讲解页。

## 问题页

1

2

IQ: 70

目标时间: 10分钟

### Q03 翻牌

这里有 100 张写着数字 1~100 的牌，并按顺序排列着。最开始所有牌都是背面朝上放置。某人从第 2 张牌开始，隔 1 张牌翻牌。然后第 2, 4, 6, ..., 100 张牌就会变成正面朝上。

接下来，另一个人从第 3 张牌开始，隔 2 张牌翻牌（原本背面朝上的，翻转成正面朝上；原本正面朝上的，翻转成背面朝上）。再接下来，又有一个人从第 4 张牌开始，隔 3 张牌翻牌（图 1）。

像这样，从第  $n$  张牌开始，每隔  $n-1$  张牌翻牌，直到没有可翻动的牌为止。

3

4

问题

求当所有牌不再变动时，所有背面朝上的牌的数字。

5

Hint!

因为只是单纯从左往右的处理，所以请用简单的方法实现。

Q03 翻牌 | 011

### 1 IQ

本书问题的难度逐章递增，每道题的 IQ 就是一个更加明确的难度提示。

### 2 目标时间

解题需要的标准思考时间。

### 3 问题的背景

为了让读者更容易理解，这里描述了问题的背景。

### 4 问题

这里是问题描述，在读者了解背景后设问，引导读者编程并解题。

### 5 提示

有助于解题的提示。

## 思路

像这样的问题用递归最容易描述。可以想象一下，切分后的木棒会像切分前的木棒一样继续被切分。如果用 Ruby，可以像代码清单 04.01 这样解决问题。

代码清单 04.01 (q04\_01.rb)

```
def cutbar(m, n, current) # current 是目前木棒的长度
  if current >= n then
    0 # 完成切分
  elsif current < m then
    1 + cutbar(m, n, current * 2) # 接下来是现在长度的 2 倍
  else
    1 + cutbar(m, n, current + m) # 加上切分次数
  end
end

puts cutbar(3, 20, 1)
puts cutbar(5, 100, 1)
```

cutbar 函数又调用了同样名称的 cutbar 函数，这是怎么回事儿呢？

像这样，函数（方式）调用函数自身就称为“递归调用”。

想要重复同样处理的时候用递归会很方便。

为了避免递归程序出错，每个递归函数都会设定一个终止条件。递归函数有个好处就是代码非常简洁。

稍稍改变一下思路会发现，还有另一个考后，本题题干可以等价于  $m$  个人黏合 1 厘米木棒。也就是说，最终使黏合的木棒总长度为 04.02 实现了这个思路。

016 | 第 1 章 入门篇

## 9 关键点

解题的关键思路。

## 10 答案

问题的答案。

## 11 专栏

讲解与该问题相关的内容，或者与编程和算法相关的内容。

## 6 下载文件名

本书中的源代码均可下载，具体信息请参考下文的“下载相关”部分。

## 7 源代码

解题的源代码示例。本书中的问题基本都会附上基于不同方法来实现的不同的源代码。

## 8 人物

本书中将有三个人物出场，一同来思考问题。

## Point

用这个方法遍历的时候可能会出现无效日期，因此需要准确排除这些日期。使用 Ruby 时，如果有错误抛出，只需要调用 rescue 就可以捕获异常。其他语言也可以利用捕获异常来简化逻辑，可以多灵活运用，即使采用同一种方法，只要精确点去优化，处理时间就能大幅度缩短。编程的时候务必要兼顾程序可读性和处理效率。

答案

19660713  
19660905  
19770217  
19950617  
20020505  
20130201

## Column

2036 年问题和 2038 年问题

大家都还记得“2000 年问题”吧？因为之前常用两位数字表示年份，所以人们认为当 2000 年来临时，根据实现逻辑不同，程序会出现各种各样的问题。比如年份的顺序会发生错乱，或者逻辑处理出错甚至直接终止运行等。不过因为事前作出了充分准备，所以并没有形成大规模的混乱。

有人说，就计算机而言，下一个日期处理问题将发生在 2036 年和 2038 年。2036 年问题的起因是有些 NTP<sup>①</sup> 协议的时间格式是以 1900 年 1 月 1 日为起点，用 32 位的二进制数表示的。2038 年问题的起因则是 C 语言等用 32 位的二进制数来表示以 1970 年 1 月 1 日为起点的时间。

这些问题有一个解决方法，就是不用 32 位，而用 64 位二进制数来表示日期。虽然距离这两个问题出现还有不少时间，但是作为开发者还是应该做到心中有数，未雨绸缪。

① NTP 是网络时间协议 (Network Time Protocol)，这是用来同步网络中各台计算机的时间的协议。——编者注

028 | 第 1 章 入门篇

## 出场人物介绍



### 吉田

在SE股份有限公司上班的年轻程序员。文科出身，偶然间撞见前辈在兴致高昂地编程，深感震撼，并立志成为工程师。好不容易才掌握了基本的编程技能，但是从学生时代起数学就是他的短板。

### 山崎

吉田的上司。在进度管理方面很严苛，但总是能耐心和大家交流。喝酒也很豪爽，深受部下尊敬。和吉田相反，山崎从小就喜欢数学，是公司“数学之美座谈会”（会员两名）的主力。



### 前辈

SE股份有限公司前员工，自由职业者。现在经常以业余活动参与者的身份出入公司，常常给公司的后辈灌输编程的乐趣。从前在公司工作时，因为超人的编程速度，留下了“手指比一般人多两根”“晚上睡觉第三只眼会睁开”的传说。

## 下载相关

本书讲解页上记载的源代码可以通过以下链接下载（点击“随书下载”）：

**URL** <http://www.ituring.com.cn/book/1814>

下载文件的著作权为作者以及出版社（翔泳社）所有。未经允许，不能通过网络传播，或者发布在Web站点上。

此外，源代码在以下执行环境中验证过：

- Ruby 2.2.3
- JavaScript 1.8
- C语言 C99 (GCC)

※本书中提到的网址等可能会随时变更。

※我们在出版本书时力求准确表述，但这不代表作者和出版社能够保证本书内容完全准确。对于因本书内容和示例运用不当导致的结果，翔泳社、原书作者、人民邮电出版社和译者均不负任何责任。

※本书中的示例程序、脚本以及运行结果页面等都是基于特定设置环境而重现的示例。

※本书中的公司名、产品名分别为各公司的商标及注册商标。

# 目录

---

## 第1章 入门篇 ★

|                 |     |
|-----------------|-----|
| 尝试用编程解决问题 ..... | 001 |
|-----------------|-----|

---

|                      |     |
|----------------------|-----|
| 二进制和十进制 .....        | 002 |
| Q01 回文十进制数 .....     | 003 |
| Q02 数列的四则运算 .....    | 007 |
| Q03 翻牌 .....         | 011 |
| Q04 切分木棒 .....       | 015 |
| Q05 还在用现金支付吗 .....   | 019 |
| Q06 (改版) 考拉兹猜想 ..... | 023 |
| Q07 日期的二进制转换 .....   | 025 |
| Q08 优秀的扫地机器人 .....   | 029 |
| Q09 落单的男女 .....      | 031 |
| Q10 轮盘的最大值 .....     | 035 |

---

## 第2章 初级篇 ★

|                     |     |
|---------------------|-----|
| 解决简单问题 体会算法效果 ..... | 039 |
|---------------------|-----|

---

|                         |     |
|-------------------------|-----|
| 性价比意识 .....             | 040 |
| Q11 斐波那契数列 .....        | 041 |
| Q12 平方根数字 .....         | 045 |
| Q13 有多少种满足字母算式的解法 ..... | 049 |
| Q14 世界杯参赛国的国名接龙 .....   | 055 |
| Q15 走楼梯 .....           | 059 |

|     |              |     |
|-----|--------------|-----|
| Q16 | 3根绳子折成四边形    | 063 |
| Q17 | 挑战30人31足     | 067 |
| Q18 | 水果酥饼日        | 071 |
| Q19 | 朋友的朋友也是朋友吗   | 075 |
| Q20 | 受难立面魔方阵      | 079 |
| Q21 | 异或运算三角形      | 083 |
| Q22 | 不缠绕的纸杯电话     | 087 |
| Q23 | 二十一点通吃       | 089 |
| Q24 | 完美的三振出局      | 091 |
| Q25 | 鞋带的时髦系法      | 093 |
| Q26 | 高效的立体停车场     | 097 |
| Q27 | 禁止右转也没关系吗    | 101 |
| Q28 | 社团活动的最优分配方案  | 105 |
| Q29 | 合成电阻的黄金分割比   | 109 |
| Q30 | 用插线板制作章鱼脚状线路 | 113 |

## 第3章 中级篇 ★★★

|             |     |
|-------------|-----|
| 优化算法 实现高速处理 | 117 |
|-------------|-----|

|                  |     |
|------------------|-----|
| 时间复杂度记法和计算量      | 118 |
| Q31 计算最短路径       | 119 |
| Q32 榻榻米的铺法       | 123 |
| Q33 飞车与角行的棋步     | 127 |
| Q34 会有几次命中注定的相遇  | 131 |
| Q35 0和7的回文数      | 133 |
| Q36 翻转骰子         | 137 |
| Q37 翻转7段码        | 143 |
| Q38 填充白色         | 149 |
| Q39 反复排序         | 153 |
| Q40 优雅的IP地址      | 157 |
| Q41 只用1个数字表示1234 | 161 |
| Q42 将牌洗为逆序       | 165 |

|     |                         |     |
|-----|-------------------------|-----|
| Q43 | 让玻璃杯水量减半 .....          | 169 |
| Q44 | 质数矩阵 .....              | 173 |
| Q45 | 排序交换次数的最少化 .....        | 177 |
| Q46 | 唯一的 $O \times$ 序列 ..... | 181 |
| Q47 | 格雷码循环 .....             | 185 |
| Q48 | 翻转得到交错排列 .....          | 189 |
| Q49 | 欲速则不达 .....             | 193 |
| Q50 | 完美洗牌 .....              | 197 |
| Q51 | 同时结束的沙漏 .....           | 201 |
| Q52 | 糖果恶作剧 .....             | 205 |
| Q53 | 同数包夹 .....              | 209 |
| Q54 | 偷懒的算盘 .....             | 213 |
| Q55 | 平分蛋糕 .....              | 217 |

## 第4章 高级篇 ★★★★★

|                    |     |
|--------------------|-----|
| 改变思路 让程序速度更快 ..... | 221 |
|--------------------|-----|

|                        |     |
|------------------------|-----|
| 编码风格 .....             | 222 |
| Q56 鬼脚图中的横线 .....      | 223 |
| Q57 最快的联络网 .....       | 229 |
| Q58 丢手绢游戏中的总移动距离 ..... | 233 |
| Q59 合并单元格的方式 .....     | 239 |
| Q60 分割为同样大小 .....      | 247 |
| Q61 不交叉,一笔画下去 .....    | 253 |
| Q62 日历的最大矩形 .....      | 259 |
| Q63 迷宫会合 .....         | 263 |
| Q64 麻烦的投接球 .....       | 269 |
| Q65 图形的一笔画 .....       | 273 |
| Q66 设计填字游戏 .....       | 279 |
| Q67 不挨着坐是一种礼节吗 .....   | 283 |
| Q68 异性相邻的座次安排 .....    | 287 |
| Q69 蓝白歌会 .....         | 291 |