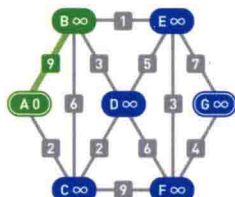
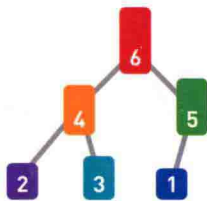
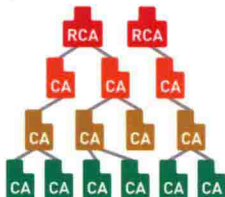


# 我的第一本 算法书

【日】石田保辉 宫崎修一 / 著 张贝 / 译



481张  
步骤图

详解26个算法  
和7个数据结构的基本原理

全彩  
印刷

 中国工信出版集团

 人民邮电出版社  
POSTS & TELECOM PRESS

# 我的第一本 算法书

[日] 石田保辉 宫崎修一 / 著 张贝 / 译

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

我的第一本算法书 / (日) 石田保辉, (日) 宫崎修  
一著; 张贝译. -- 北京: 人民邮电出版社, 2018.11

(图灵程序设计丛书)

ISBN 978-7-115-49524-2

I. ①我… II. ①石… ②宫… ③张… III. ①电子计  
算机—算法理论 IV. ①TP301.6

中国版本图书馆CIP数据核字(2018)第223107号

## 内 容 提 要

本书采用大量图片, 通过详细的分步讲解, 以直观、易懂的方式展现了7个数据结构和26个基础算法的基本原理。第1章介绍了链表、数组、栈等7个数据结构; 从第2章到第7章, 分别介绍了和排序、查找、图论、安全、聚类等相关的26个基础算法, 内容涉及冒泡排序、二分查找、广度优先搜索、哈希函数、迪菲-赫尔曼密钥交换、*k*-means 算法等。

本书没有枯燥的理论和复杂的公式, 而是通过大量的步骤图帮助读者加深对数据结构原理和算法执行过程的理解, 便于学习和记忆。将本书作为算法入门的第一步, 是非常不错的选择。

本书适合所有对算法感兴趣, 想要从零开始学习算法的读者阅读。

- 
- ◆ 著 [日] 石田保辉 宫崎修一  
译 张 贝  
责任编辑 高宇涵  
责任印制 周昇亮
- ◆ 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>  
北京捷迅佳彩印刷有限公司印刷
- ◆ 开本: 880×1230 1/24  
印张: 8.5  
字数: 210千字 2018年11月第1版  
印数: 1-4 000册 2018年11月北京第1次印刷
- 著作权合同登记号 图字: 01-2017-7988号
- 

定价: 69.00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](http://iTuring.cn)

## 版权声明

アルゴリズム図鑑

(Algorithm Zukan : 4977-6)

Copyright© 2017 by Moriteru Ishida, Shuichi Miyazaki.

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 © 2018 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。

## 本书主页

<http://www.ituring.com.cn/book/2464>

阅读本书后，您可通过以上网址将您的感想、意见或问题写下来。

另外，点击页面右侧的“提交 / 查看勘误”可以提交或查看本书勘误。

※ 本书中的 URL 等信息可能会在未予通知的情况下发生变化。

※ 本书在出版之际已力求内容的准确性，但是对于应用本书内容和示例所产生的一切后果，本书作者、翔泳社、人民邮电出版社和译者概不负责。

※ 本书中提及的公司名和商品名皆是各相关公司的商标或注册商标。



## 前言

本书以 iOS 和 Android 平台上的应用程序“算法动画图解”为基础，以图配文，详细讲解了各种算法和数据结构的基本原理。如果本书能够帮助大家理解基本算法的操作和特征，那么我将感到十分荣幸。

使用不同的算法解决同一个问题时，就算得到的结果是一样的，算法之间的性质也有很大的差异。比如，某个算法的运行时间很短，但需要占用大量内存；而另一个算法运行时间较长，但内存资源占用较少。学习各种算法可以使我们在编程时有更多的选择。成为优秀程序员的必要条件之一，就是可以根据应用场景选择最合适的算法。

如果您对算法有兴趣，还可以挑战一下“算法理论”这门学科，试着去发现更高效的算法，或者研究目前用算法还无法解决的问题。

石田保辉

算法是解决问题的计算步骤，用于编写程序之前。即使是解决同样的问题，高效算法和低效算法所花费的时间也迥然不同。另外，要想执行高效的算法，还需要使用合适的数据结构。本书的目的就是让初学者也能轻松地理解算法和数据结构。

本书以 iOS 和 Android 平台上的应用程序“算法动画图解”为基础。该应用以动画的形式展示了算法的流程，而本书则采用了大量的图片来分步讲解，尽量保留了原应用易懂的优点。为了配合出版，本书还添加了“什么是算法”“算法的运行时间”“图的基础知识”等应用中没有的章节，相信会让读者对算法的理解更加深刻。

读完本书，不过是站在了算法世界的入口，这个世界还有很多领域等待人们去探索。如果您由此对算法产生了兴趣，请务必继续深入学习。

宫崎修一

## 谢辞

本书中大量使用了应用程序“算法动画图解”中的图片，在使用之前，我们得到了图片制作者光森裕树先生的许可。此外，从选题策划、内容编辑到出版进度管理，翔泳社的秦和宏先生在本书的整个出版流程中都付出了颇多心血。在此对二位表示由衷的感谢。

石田保辉 宫崎修一

# 关于应用程序“算法动画图解”的说明

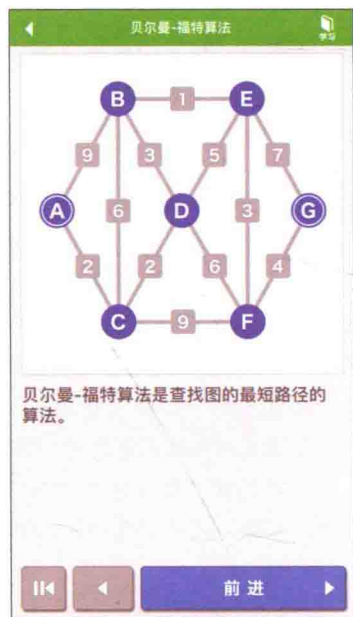
本书根据 iOS 和 Android 平台上的应用程序“算法动画图解”编写而成，为配合图书出版，对内容进行了补充和修正，专门添加了基础理论方面的内容。

对于本书中出现的各个算法，应用内都采用动画交互的方式进行了讲解，部分算法还可以通过改变设置尝试不同的模式。将应用和本书结合使用，可以加深对相关知识的理解。请各位读者通过下面的步骤下载该应用并加以灵活运用。



## ► iPhone / iPad 用户

- ① 打开 App Store
- ② 点击“搜索”，输入“算法动画图解”
- ③ 点击“获取”



## ► Android 用户

- ① 打开 Google Play
- ② 打开页面上的“搜索”项，输入“算法动画图解”进行搜索
- ③ 进入应用“Algorithms: Explained and Animated”的页面，点击“安装”

※ 应用为免费下载，可以免费试用部分算法，但解锁所有项目需要在应用内购买。



---

# Contents | 目录

序章	<b>算法的基本知识</b> .....	001
	0-1 什么是算法 .....	002
	0-2 运行时间的计算方法 .....	007

第	<b>1</b>	章	<b>数据结构</b> .....	009
	1-1		什么是数据结构 .....	010
	1-2		链表 .....	013
	1-3		数组 .....	016
	1-4		栈 .....	020
	1-5		队列 .....	022
	1-6		哈希表 .....	024
	1-7		堆 .....	032
	1-8		二叉查找树 .....	036

---

第 **2** 章 **排序** ..... 043

2-1 什么是排序 ..... 044

2-2 冒泡排序 ..... 046

2-3 选择排序 ..... 050

2-4 插入排序 ..... 052

2-5 堆排序 ..... 056

2-6 归并排序 ..... 060

2-7 快速排序 ..... 064

第 **3** 章 **数组的查找** ..... 071

3-1 线性查找 ..... 072

3-2 二分查找 ..... 074

第 **4** 章 **图的搜索** ..... 077

4-1 什么是图 ..... 078

4-2 广度优先搜索 ..... 082

4-3 深度优先搜索 ..... 086

4-4	贝尔曼 - 福特算法 .....	090
4-5	狄克斯特拉算法 .....	096
4-6	A* 算法 .....	103

## 第 5 章 安全算法 ..... 107

5-1	安全和算法 .....	108
5-2	加密的基础知识 .....	112
5-3	哈希函数 .....	116
5-4	共享密钥加密 .....	120
5-5	公开密钥加密 .....	124
5-6	混合加密 .....	132
5-7	迪菲 - 赫尔曼密钥交换 .....	136
5-8	消息认证码 .....	144
5-9	数字签名 .....	152
5-10	数字证书 .....	158

## 第 6 章 聚类 ..... 165

6-1	什么是聚类 .....	166
6-2	k-means 算法 .....	168

---

第 **7** 章 其他算法 ..... 173

7-1 欧几里得算法 ..... 174

7-2 素性测试 ..... 178

7-3 网页排名 ..... 182

7-4 汉诺塔 ..... 190

# 序章

---

## 算法的基本知识



No.

0-1

# 什么是算法

## 算法与程序的区别

算法就是计算或者解决问题的步骤。我们可以把它想象成食谱。要想做出特定的料理，就要遵循食谱上的步骤；同理，要想用计算机解决特定的问题，就要遵循算法。这里所说的特定问题多种多样，比如“将随意排列的数字按从小到大的顺序重新排列”“寻找出发点到目的地的最短路径”，等等。

食谱和算法之间最大的区别就在于算法是严密的。食谱上经常会有描述得比较模糊的部分，而算法的步骤都是用数学方式来描述的，所以十分明确。

算法和程序有些相似，区别在于程序是以计算机能够理解的编程语言编写而成的，可以在计算机上运行，而算法是以人类能够理解的方式描述的，用于编写程序之前。不过，在这个过程中到哪里为止是算法、从哪里开始是程序，并没有明确的界限。

就算使用同一个算法，编程语言不同，写出来的程序也不同；即便使用相同的编程语言，写程序的人不同，那么写出来的程序也是不同的。

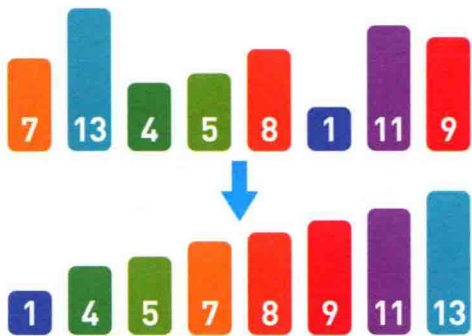
## 排列整数的算法：排序

### ▶ 查找最小的数字并交换：选择排序

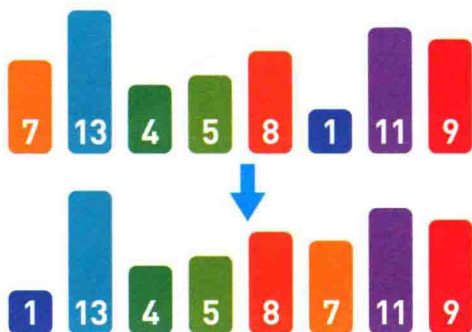
来看一个具体的算法示例吧。这是一个以随意排列的整数为输入，把它们按从小到大的顺序重新排列的问题。这类排序问题我们将在第2章详细讲解。

只解决这一个问题很简单，但是算法是可以应对任意输入的计算步骤，所以必须采用通用的描述。虽然在这个示例中输入的整数个数  $n$  为 8，然而不管  $n$  多大，算法都必须将问题解决。

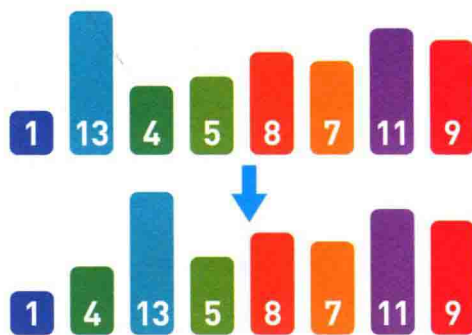
那么，你首先想到的方法，是不是先从输入的数字中找出最小的数字，再将它和最左边的数字交



换位置呢？在这个示例中就是找到最小数字 1，然后将它和最左边的 7 交换位置。



这之后 1 的位置便固定下来，不再移动。接下来，在剩下的数字里继续寻找最小数，再将它和左边第 2 个数字交换位置。于是，4 和 13 也交换了位置。



我们将这样的一次交换称为“1 轮”。到了第  $k$  轮的时候，就把剩下的数字中最小的一个，与左边开始第  $k$  个数字进行交换。于是在结束第  $k$  轮后，从左数的  $k$  个数字便都按从小到大的顺序排列了。只要将这个步骤重复  $n$  次，那么所有的数字都将按从小到大的顺序排列。

这便是我们将在 2-3 节中介绍的选择排序。不管输入的数字是什么、 $n$  有多大，都可以用这个算法解决问题。

### ► 用计算机能理解的方式构思解法：算法的设计

计算机擅长高速执行一些基本命令，但无法执行复杂的命令。此处的“基本命令”指的是“做加法”或者“在指定的内存地址上保存数据”等。

计算机是以这些基本命令的组合为基础运行的，面对复杂的操作，也是通过搭配组合这些

基本命令来应对的。上文中提到的“对  $n$  个数字进行排序”对计算机来说就是复杂的操作。如何设计算法来解决这个排序问题，也就等同于构思如何搭配组合计算机可以执行的那些基本命令来实现这个操作。

## 如何选择算法

能解决排序问题的算法不止选择排序这一个。那么，当有多个算法都可以解决同一个问题时，我们该如何选择呢？在算法的评判上，考量的标准也各有不同。

比如，简单的算法对人来说易于理解，也容易被写成程序，而在运行过程中不需要耗费太多空间资源的算法，就十分适用于内存小的计算机。

不过，一般来说我们最为重视的是算法的运行时间，即从输入数据到输出结果这个过程所花费的时间。

## 对 50 个数字排序所花的时间竟然比宇宙的历史还要长吗

### ▶ 使用全排列算法进行排序

为了让大家体会一下低效率算法的效果，这里来看看下面这个排序算法。

- ① 生成一个由  $n$  个数字构成的数列（不和前面生成的数列重复）
- ② 如果①中生成的数列按从小到大的顺序排列就将其输出，否则回到步骤①

我们就把这个算法称为“全排列算法”吧。全排列算法列出了所有的排列方法，所以不管输入如何，都可以得到正确的结果。

那么，需要等多久才能出结果呢？若运气好，很快就能出现正确排列的话，结果也就立马出来了。然而，实际情况往往并不如我们所愿。最差的情况，也就是直到最后才出现正确排列的情况下，计算机就不得不确认所有可能的排列了。

$n$  个数字有  $n!$  种不同的排列方法 ( $n! = n(n-1)(n-2)\cdots 3 \cdot 2 \cdot 1$ )。现在，我们来看看  $n=50$  时是怎样一种情况吧。

- ①  $50! = 50 \cdot 49 \cdot 48 \cdots 3 \cdot 2 \cdot 1$
- ②  $50 \cdot 49 \cdot 48 \cdots 3 \cdot 2 \cdot 1 > 50 \cdot 49 \cdot 48 \cdots 13 \cdot 12 \cdot 11$
- ③  $50 \cdot 49 \cdot 48 \cdots 13 \cdot 12 \cdot 11 > 10^{40}$

公式①中， $50!$ 即为数字1到数字50的乘积。为了便于计算，我们通过公式②③将结果近似转换为10的 $n$ 次方的形式。公式②右边部分去掉了10以下的数字，因此小于 $50!$ 。公式③左右都是40个数字的乘积，但左边数字都大于10，因此大于右边的 $10^{40}$ 。接下来我们就用 $10^{40}$ 近似代表50个数字的所有排列情况来进行计算。

假设1台高性能计算机1秒能检查1万亿( $=10^{12}$ )个数列，那么检查 $10^{40}$ 个数列将花费的时间为 $10^{40} \div 10^{12} = 10^{28}$ 秒。1年为31 536 000秒，不到 $10^8$ 秒。因此， $10^{28}$ 秒 $> 10^{20}$ 年。

从大爆炸开始宇宙已经经历了约137亿年，即便如此也少于 $10^{11}$ 年。也就是说，仅仅是对50个数字进行排序，若使用全排列算法，就算花费宇宙年龄的 $10^9$ 倍时间也得不出答案。

