

书中所有代码读者均可到 <http://www.tdpress.com/zyzx/tsscflwj> 下载使用

Java



常用算法手册

The Classic Algorithm Fundamental of
Program Language Java

算法是一切程序设计的**基础**和**灵魂**，更是一位程序员水平高低的**集中体现**。

● 赵志云 衡友跃 编著

涵盖广泛 精炼的理论讲述搭配大量经典算法示例，学习查询兼而有之。

阐述到位 算法思想、算法实现和完整示例合理搭配，相辅相成。

示例完善 示例分析精准，代码注释精确，每段代码皆可通过编译执行。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Java

常用算法手册

**The Classic Algorithm Fundamental of
Program Language Java**

● 赵志云 衡友跃 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

现代的设计任务大多通过计算机编程来完成,而算法起到了至关重要的作用。可以毫不夸张地说,算法是一切程序设计的灵魂和基础。选择合理的算法,可以起到事半功倍的效果。

本书分三篇,共13章,分别介绍了算法基础、算法应用和算法面试题。首先介绍了算法概述,然后重点分析了数据结构和基本算法思想;接着,详细讲解了算法在排序、查找、数学计算、数论、历史趣题、游戏、密码学等领域中的应用;最后,列举了算法的一些常见面试题。

书中知识点覆盖全面,结构安排紧凑,讲解详细,实例丰富。全书对每一个知识点都给出了相应的算法及应用实例,虽然这些例子都是以Java语言来编写的,但是这些算法并不局限于Java语言。如果读者采用其他的编程语言,例如C++、C、C#等,只要根据其语法格式进行适当的修改便可以了。

本书旨在帮助Java语言初级程序员深入了解Java算法思想,提升其语言代码编程能力,还可以作为大中专院校学生学习数据结构和算法的参考书,也可作为教师授课提供素材,同时还可以作为具有一定编程经验的程序员的参考用书。

图书在版编目(CIP)数据

Java 常用算法手册 / 赵志云, 衡友跃编著. -- 北京:
中国铁道出版社, 2012. 5
ISBN 978-7-113-14366-4

I. ①J… II. ①赵… ②衡… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第034739号

书 名: Java 常用算法手册
作 者: 赵志云 衡友跃 编著

策划编辑: 荆 波
责任编辑: 荆 波
责任印制: 赵星辰

读者服务热线: 010-63560056
编辑助理: 张 丹

出版发行: 中国铁道出版社(北京市西城区右安门西街8号 邮政编码: 100054)
印 刷: 北京鑫正大印刷有限公司
版 次: 2012年5月第1版 2012年5月第1次印刷
开 本: 787mm×1092mm 1/16 印张: 27 字数: 656千
书 号: ISBN 978-7-113-14366-4
定 价: 59.00元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社发行部联系调换。

信息社会进入云计算时代，最为明显的特征就是“各种云终端+云服务器应用的组合”。不管是编写服务器端的程序，还是编写 PC、平板电脑、手机等云终端上的应用程序，采用 Java 基本上都是最常见的选择。而一个应用程序往往由编程语言、数据结构和算法来组成。其中，算法是整个程序设计的核心。算法代表着求解具体问题的手段和方法，可以毫不夸张地说，算法是一切程序设计的灵魂和基础。选择合理的算法，可以起到事半功倍的效果。因此，对于程序员来说，学习和掌握算法成为重中之重。同时，各大公司招聘 Java 程序员时，除基本语法之外，算法的掌握程度也是考核的重点方面。

本书特色

为了保证读者掌握算法这个程序设计的核心技术，我们一开始就为本书规划了一些特色，以保证它的质量和生命力。和其他书籍相比，本书有如下优点：

(1) 本书由浅入深，循序渐进地带领读者逐步深入学习算法和数据结构的知识。

(2) 本书不仅详细介绍了算法的基础知识和应用，还对常用的加密解密和数据压缩等高级算法编程技术进行了详细的讲解。

(3) 本书在讲解每个知识点的同时，均给出了相应的算法原理、算法实现，同时还给出了完整的实例，每个实例都可以运行，使得读者可以快速掌握对应知识点如何应用在程序设计中。

(4) 本书在介绍各个知识点的时候，尽量结合历史背景并给出了问题的完整分析，使读者可以了解问题的来龙去脉，避免了代码类书籍的枯燥乏味。

(5) 本书对每一个实例的程序代码都进行了详细的注释和分析，并给出了运行结果，使得读者更加容易理解。

(6) 本书中的所有代码均采用应用较为广泛的 Java 语言进行编写。但是这些算法本身并不仅限于 Java 语言，读者如果采用 C++、C、C#、VB 等其他编程语言，只需按照对应的语法格式进行少量的修改便可以使用。

本书的内容

本书以实用性、系统性、完整性和前沿性为重点，详细介绍了算法的基本思想和在不同领域的应用实例。本书分为三篇，共 13 章内容。

第一篇 算法基础篇

本篇共 3 章，详细介绍了算法和数据结构的相关知识。

第 1 章 算法概述，本篇介绍了什么是算法，以及算法的发展、分类、算法的基本概念、算法的表示以及性能评价等。同时，本章还通过一个实例分析算法的编写、编译执行等。最后，本章还介绍了一些算法的最新进展，作为读者扩展知识面。

第 2 章为数据结构，首先介绍了数据结构的基本概念，然后重点介绍了顺序表结构、链表结构、栈结构、队列结构、树结构和图结构的实现。

第3章为基本算法思想，介绍了穷举算法思想、递推算法思想、递归算法思想、分治算法思想和概率算法思想的原理和应用。

第二篇 算法应用篇

本篇共9章，详细讲解了算法在排序、查找、数学计算、数论、历史趣题和游戏中的应用。

第4章为排序算法，详细介绍各种排序算法及其应用实例。

第5章为查找算法，详细讲解了各种查找算法及其应用实例。

第6章为基本数学问题，介绍了多项式计算、随机数生成、复数运算、阶乘、计算圆周率、矩阵运算和方程求解中的算法，并通过完整的例子来演示了算法的应用。

第7章为数据结构问题，介绍与数据结构相关的一些典型问题，包括动态数组排序、约瑟夫环、最短总距离、最短路径和括号匹配等。

第8章为数论问题，首先介绍了数论的基本知识，然后介绍了初等数论中的一些基本数据的计算算法。

第9章为算法经典趣题，介绍了一些历史上非常有名的问题、智力趣题等。这些经典趣题是历史上累积的算法财富，通过这些有趣的问题，可以提高读者的编程能力和学习算法的兴趣。

第10章为游戏中的算法，讲解了一些常用游戏中的算法，包括洗扑克牌算法、取火柴游戏算法、十点半算法以及生命游戏算法。

第11章为密码学算法，首先介绍了密码学的基本知识，然后详细介绍了换位加密算法、替换加密算法、位加密算法和一次一密加密算法。

第12章为压缩与解压缩算法，首先介绍了数据压缩的基本知识，然后重点介绍了LZW压缩算法和解压缩算法的实现，最后结合实例讲解了压缩算法的应用。

第三篇 算法面试篇

本篇共1章，详细讲解了常见面试中的算法。

本书结构紧凑，知识点涉及全面，内容翔实，案例丰富。由于本书内容较多，编写时间较仓促，书中如有遗漏或不足之处，恳请广大读者提出宝贵意见，发便改进。

适合的读者

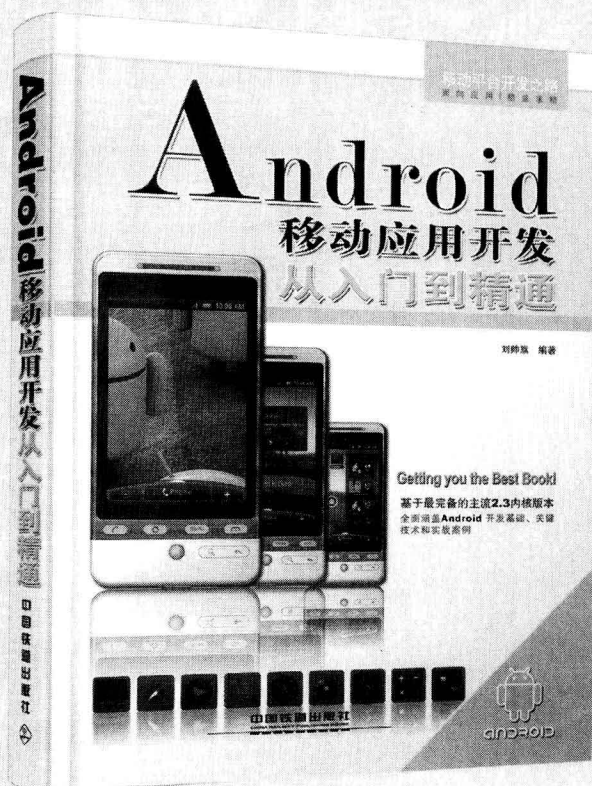
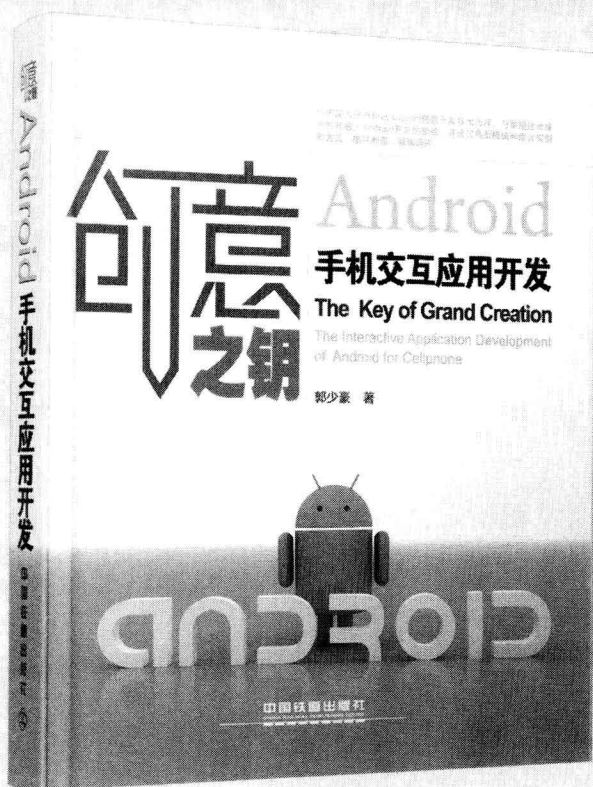
- 大专院校相关专业的学生及教师；
- 系统开发人员；
- 程序设计初学者；
- Java程序员；
- 计算机程序设计爱好者。

阅读完本书，读者若需要进一步提高和学习其他算法的相关技术，也可以到读者服务网站：<http://www.rzchina.net/forum.php>上阅读在实际工作中可能会用到的其他算法技术资料。同时，我们还会在这个网站上提供辅助的教学资料，供读者下载或者在线学习。

编者

2012年3月

定位于最主流、最完备的Android内核开发版本，搭建合理完善的知识架构，将开发基础、关键技术、典型应用和综合实例完美融合，巧妙融入作者多年开发经验，引领读者拾级而上，全面掌握Android开发技术。



ANDROID

第 1 篇 算法基础篇

第 1 章 算法概述

1.1 什么是算法	1	2.1.6 数据类型	19
1.2 算法的发展历史	2	2.1.7 常用的数据结构	20
1.3 算法的分类	3	2.1.8 选择合适的数据结构解决 实际问题	21
1.4 算法相关概念的区别	3	2.2 线性表	21
1.4.1 算法与公式的关系	4	2.2.1 什么是线性表	21
1.4.2 算法与程序的关系	4	2.2.2 线性表的基本运算	22
1.4.3 算法与数据结构的关系	4	2.3 顺序表结构	23
1.5 算法的表示	4	2.3.1 准备数据	23
1.5.1 自然语言表示	5	2.3.2 初始化顺序表	24
1.5.2 流程图表示	5	2.3.3 计算顺序表长度	24
1.5.3 N-S 图表示	6	2.3.4 插入结点	24
1.5.4 伪代码表示	7	2.3.5 追加结点	25
1.6 算法的性能评价	7	2.3.6 删除结点	25
1.6.1 时间复杂度	8	2.3.7 查找结点	25
1.6.2 空间复杂度	8	2.3.8 显示所有结点	26
1.7 算法实例	8	2.3.9 顺序表操作实例	26
1.7.1 查找数字	8	2.4 链表结构	30
1.7.2 创建项目	10	2.4.1 什么是链表结构	30
1.7.3 编译执行	12	2.4.2 准备数据	31
1.8 算法的新进展	13	2.4.3 追加结点	31
1.9 小结	14	2.4.4 插入头结点	32
		2.4.5 查找结点	33
		2.4.6 插入结点	34
		2.4.7 删除结点	35
		2.4.8 计算链表长度	35
		2.4.9 显示所有结点	36
		2.4.10 链表操作实例	36
		2.5 栈结构	41
		2.5.1 什么是栈结构	41
		2.5.2 准备数据	41

第 2 章 数据结构

2.1 数据结构概述	15
2.1.1 什么是数据结构	15
2.1.2 数据结构中的基本概念	16
2.1.3 数据结构的内容	16
2.1.4 数据结构的分类	18
2.1.5 数据结构的几种存储方式	18

2.5.3	初始化栈结构	42
2.5.4	判断空栈	42
2.5.5	判断满栈	43
2.5.6	清空栈	43
2.5.7	释放空间	43
2.5.8	入栈	43
2.5.9	出栈	44
2.5.10	读结点数据	44
2.5.11	栈结构操作实例	45
2.6	队列结构	47
2.6.1	什么是队列结构	48
2.6.2	准备数据	48
2.6.3	初始化队列结构	49
2.6.4	判断空队列	49
2.6.5	判断满队列	49
2.6.6	清空队列	50
2.6.7	释放空间	50
2.6.8	入队列	50
2.6.9	出队列	51
2.6.10	读结点数据	51
2.6.11	计算队列长度	52
2.6.12	队列结构操作实例	52
2.7	树结构	55
2.7.1	什么是树结构	56
2.7.2	树的基本概念	56
2.7.3	二叉树	57
2.7.4	准备数据	60
2.7.5	初始化二叉树	61
2.7.6	添加结点	61
2.7.7	查找结点	63
2.7.8	获取左子树	63
2.7.9	获取右子树	64
2.7.10	判断空树	64
2.7.11	计算二叉树深度	64

2.7.12	清空二叉树	65
2.7.13	显示结点数据	65
2.7.14	遍历二叉树	65
2.7.15	树结构操作实例	67
2.8	图结构	74
2.8.1	什么是图结构	74
2.8.2	图的基本概念	75
2.8.3	准备数据	79
2.8.4	创建图	81
2.8.5	清空图	82
2.8.6	显示图	82
2.8.7	遍历图	83
2.8.8	图结构操作实例	84
2.9	小结	87

第 3 章 基本算法思想

3.1	常用算法思想概述	88
3.2	穷举算法思想	88
3.2.1	穷举算法基本思想	89
3.2.2	穷举算法实例	89
3.3	递推算法思想	90
3.3.1	递推算法基本思想	91
3.3.2	递推算法实例	91
3.4	递归算法思想	92
3.4.1	递归算法基本思想	93
3.4.2	递归算法实例	93
3.5	分治算法思想	94
3.5.1	分治算法基本思想	94
3.5.2	分治算法实例	95
3.6	概率算法思想	98
3.6.1	概率算法基本思想	99
3.6.2	概率算法实例	99
3.7	小结	101

第 2 篇 算法应用篇

第 4 章 排序算法

4.1	排序算法概述	102
4.2	冒泡排序法	103
4.2.1	冒泡排序算法	103
4.2.2	冒泡排序算法实例	104

4.3	选择排序法	106
4.3.1	选择排序算法	106
4.3.2	选择排序算法实例	107
4.4	插入排序法	109
4.4.1	插入排序算法	109
4.4.2	插入排序算法实例	111
4.5	Shell 排序法	112
4.5.1	Shell 排序算法	112
4.5.2	Shell 排序算法实例	113
4.6	快速排序法	115
4.6.1	快速排序算法	115
4.6.2	快速排序算法实例	117
4.7	堆排序法	119
4.7.1	堆排序算法	119
4.7.2	堆排序算法实例	123
4.8	合并排序法	125
4.8.1	合并排序算法	126
4.8.2	合并排序算法实例	128
4.9	排序算法的效率	131
4.10	排序算法的其他应用	132
4.10.1	反序排序	132
4.10.2	字符串数组的排序	134
4.10.3	字符串的排序	137
4.11	小结	139

第 5 章 查找算法

5.1	查找算法概述	140
5.2	顺序查找	141
5.2.1	顺序查找算法	141
5.2.2	顺序查找操作实例	141
5.3	折半查找	143
5.3.1	折半查找算法	143
5.3.2	折半查找操作实例	145
5.4	数据结构中的查找算法	147
5.4.1	顺序表结构中的查找算法	148
5.4.2	链表结构中的查找算法	151
5.4.3	树结构中的查找算法	154
5.4.4	图结构中的查找算法	155
5.5	小结	156

第 6 章 基本数学问题

6.1	判断闰年	157
6.2	多项式计算	159
6.2.1	一维多项式求值	159
6.2.2	二维多项式求值	161
6.2.3	多项式乘法	163
6.2.4	多项式除法	164
6.3	随机数生成算法	167
6.3.1	Java 语言中的随机方法	167
6.3.2	[0, 1]之间均匀分布的 随机数算法	169
6.3.3	产生任意范围的随机数	170
6.3.4	[m, n]之间均匀分布的随机 整数算法	171
6.3.5	正态分布的随机数生成算法	173
6.4	复数运算	174
6.4.1	简单的复数运算	175
6.4.2	复数的幂运算	177
6.4.3	复指数运算	178
6.4.4	复对数运算	180
6.4.5	复正弦运算	181
6.4.6	复余弦运算	182
6.5	阶乘	183
6.5.1	使用循环来计算阶乘	183
6.5.2	使用递归来计算阶乘	184
6.6	计算 π 的近似值	185
6.6.1	割圆术	186
6.6.2	蒙特卡罗算法	189
6.6.3	级数公式	191
6.7	矩阵运算	193
6.7.1	矩阵加法	193
6.7.2	矩阵减法	195
6.7.3	矩阵乘法	196
6.8	方程求解	198
6.8.1	线性方程求解——高斯 消元法	198
6.8.2	非线性方程求解—— 二分法	203

6.8.3 非线性方程求解——牛顿迭代法	205
6.9 小结	208

第 7 章 数据结构问题

7.1 动态数组排序	209
7.1.1 动态数组的存储和排序	209
7.1.2 动态数组排序实例	210
7.2 约瑟夫环	212
7.2.1 简单约瑟夫环算法	213
7.2.2 简单约瑟夫环求解	214
7.2.3 复杂约瑟夫环算法	216
7.2.4 复杂约瑟夫环求解	217
7.3 城市之间的最短总距离	220
7.3.1 最短总距离算法	220
7.3.2 最短总距离求解	222
7.4 最短路径	227
7.4.1 最短路径算法	227
7.4.2 最短路径求解	229
7.5 括号匹配	234
7.5.1 括号匹配算法	235
7.5.2 括号匹配求解	236
7.6 小结	239

第 8 章 数论问题

8.1 数论概述	240
8.1.1 数论概述	240
8.1.2 数论的分类	241
8.1.3 初等数论	242
8.1.4 本章用到的基本概念	242
8.2 完全数	243
8.2.1 什么是完全数	243
8.2.2 计算完全数算法	244
8.3 亲密数	246
8.3.1 什么是亲密数	246
8.3.2 计算亲密数算法	246
8.4 水仙花数	249
8.4.1 什么是水仙花数	249
8.4.2 计算水仙花数算法	250

8.5 自守数	252
8.5.1 什么是自守数	252
8.5.2 计算自守数算法	253
8.6 最大公约数	257
8.6.1 计算最大公约数算法——辗转相除法	257
8.6.2 计算最大公约数算法——Stein 算法	258
8.6.3 计算最大公约数示例	259
8.7 最小公倍数	260
8.8 素数	262
8.8.1 什么是素数	262
8.8.2 计算素数算法	262
8.9 回文素数	264
8.9.1 什么是回文素数	264
8.9.2 计算回文素数算法	264
8.10 平方回文数	267
8.10.1 什么是平方回文数	267
8.10.2 计算平方回文数算法	267
8.11 分解质因数	269
8.12 小结	271

第 9 章 算法经典趣题

9.1 百钱买百鸡	272
9.1.1 百钱买百鸡算法	272
9.1.2 百钱买百鸡求解	273
9.2 五家共井	274
9.2.1 五家共井算法	274
9.2.2 五家共井求解	275
9.3 鸡兔同笼	277
9.3.1 鸡兔同笼算法	277
9.3.2 鸡兔同笼求解	277
9.4 猴子吃桃	278
9.4.1 猴子吃桃算法	278
9.4.2 猴子吃桃求解	279
9.5 舍罕王赏麦	280
9.5.1 舍罕王赏麦问题	280
9.5.2 舍罕王赏麦求解	281
9.6 汉诺塔	282

9.6.1	汉诺塔算法	282
9.6.2	汉诺塔求解	284
9.7	窃贼问题	285
9.7.1	窃贼问题算法	285
9.7.2	窃贼问题求解	287
9.8	马踏棋盘	290
9.8.1	马踏棋盘算法	290
9.8.2	马踏棋盘求解	291
9.9	八皇后问题	294
9.9.1	八皇后问题算法	294
9.9.2	八皇后问题求解	295
9.10	寻找假银币	297
9.10.1	寻找假银币算法	297
9.10.2	寻找假银币求解	299
9.11	青蛙过河	301
9.11.1	青蛙过河算法	302
9.11.2	青蛙过河求解	303
9.12	三色旗	306
9.12.1	三色旗算法	306
9.12.2	三色旗求解	308
9.13	渔父捕鱼	310
9.13.1	渔父捕鱼算法	310
9.13.2	渔父捕鱼求解	310
9.14	爱因斯坦的阶梯	311
9.14.1	爱因斯坦的阶梯算法	312
9.14.2	爱因斯坦的阶梯求解	312
9.15	兔子产仔	313
9.15.1	兔子产仔算法	313
9.15.2	兔子产仔求解	314
9.16	常胜将军	315
9.16.1	常胜将军算法	315
9.16.2	常胜将军求解	316
9.17	新郎和新娘	318
9.17.1	新郎和新娘算法	318
9.17.2	新郎和新娘求解	319
9.18	三色球	320
9.18.1	三色球算法	320
9.18.2	三色球求解	321

9.19	小结	322
------	----	-----

第 10 章 游戏中的算法

10.1	洗扑克牌算法	323
10.1.1	洗扑克牌算法	323
10.1.2	洗扑克牌实例	324
10.2	取火柴游戏算法	327
10.2.1	取火柴游戏算法	327
10.2.2	取火柴游戏实例	328
10.3	10 点半算法	330
10.3.1	10 点半算法	330
10.3.2	10 点半游戏实例	335
10.4	生命游戏	342
10.4.1	生命游戏的原理	342
10.4.2	生命游戏的算法	343
10.4.3	生命游戏实例	344
10.5	小结	349

第 11 章 密码学概述

11.1	密码学概述	350
11.1.1	密码学的发展	350
11.1.2	密码学的基本概念	351
11.1.3	柯克霍夫斯原则	352
11.1.4	经典密码学算法	352
11.2	换位加密解密算法	353
11.2.1	换位加密解密算法	353
11.2.2	换位加密解密算法实例	356
11.3	替换加密解密算法	359
11.3.1	替换加密解密算法	359
11.3.2	替换加密解密算法实例	360
11.4	位加密解密算法	363
11.4.1	位加密解密算法	363
11.4.2	位加密解密算法实例	364
11.5	一次一密加密解密算法	366
11.5.1	一次一密加密解密算法	366
11.5.2	一次一密加密解密算法实例	367
11.6	小结	369

第 12 章 压缩与解压缩算法

12.1 压缩与解压缩概述.....	370	12.2 压缩算法.....	371
12.1.1 压缩与解压缩分类.....	370	12.3 解压缩算法.....	374
12.1.2 典型的压缩解压缩算法.....	370	12.4 压缩解压缩实例.....	376
		12.5 小结.....	384

第 3 篇 算法面试篇

第 13 章 算法面试题

13.1 基础算法.....	385	13.2 思维扩展算法.....	403
13.1.1 字符串匹配.....	385	13.2.1 蛇形打印.....	403
13.1.2 哥德巴赫猜想的近似证明...	388	13.2.2 24 点算法.....	405
13.1.3 将一个正整数分解质因数...	389	13.2.3 双色球随机摇号.....	409
13.1.4 怎样实现金额转换.....	391	13.2.4 巧妙过桥.....	412
13.1.5 数字排列.....	396	13.2.5 猴子吃桃.....	415
13.1.6 数字拆解.....	398	13.2.6 天平称物.....	416
13.1.7 数字组合.....	400	13.2.7 掷骰子游戏.....	418
		13.3 小结.....	421

第 1 章 算法概述

计算机技术，特别是计算机程序设计大大改变了人们的工作方式。现代的设计任务大多通过计算机编程交给计算机来完成。其中，算法起到了至关重要的作用。可以毫不夸张地说，算法是一切程序设计的灵魂和基础。本章介绍算法的一些基本概念、发展历史、算法表示和应用等。

1.1 什么是算法

那么究竟什么是算法（algorithm）呢？从字面意义上也比较好理解，算法也就是用于计算的方法，通过这种方法可以达到预期的计算结果。

此外，在一般的教科书或者字典上也有关于算法的专业解释，例如：算法是解决实际问题的一种精确描述方法、算法是对特定问题的求解步骤的一种精确描述方法等。目前，被广泛认可的算法的专业定义是，算法是模型分析的一组可行的、确定的和有穷的规则。

其实，通俗地讲，算法可以理解为一个完整的解题步骤，由一些基本运算和规定的运算顺序而构成。通过这样的解题步骤可以解决特定的问题。从计算机程序设计的角度看，算法由一系列求解问题的指令构成，能够根据规范的输入，在有限的时间内获得有效的输出结果。算法代表了用系统的方法来描述解决问题的一种策略机制。

我们可以举一个例子来看看算法是如何在现实生活中发挥作用的。最典型的例子就是统筹安排，假设我们要有 3 件事（事件 A、事件 B 和事件 C）要做：

- 做事件 A 需要耗费 5 分钟；
- 做事件 B 需要耗费 5 分钟但需要 15 分钟的时间才可以得到结果，例如烧水等待水开的过程；
- 做事件 C 需要耗费 10 分钟。

那么我们应该怎么来做这三件事情呢？一种方法是依次做，如图 1-1 所示，做完事件 A，再做事情 B，最后做事情 C。这样，总的耗时是 $5 + (5 + 15) + 10 = 35$ 分钟，这显然是浪费时间的一种方法。

在实际生活中比较可取的方法是，先做事件 B，在等待事件 B 完成的过程中做事件 A 和事件 C。这样，等待事件 B 完成的 15 分钟正好可以完成事件 A 和事件 C。此时，总的耗时是 $5 + 15 = 20$ 分钟，效率明显提高，如图 1-2 所示。

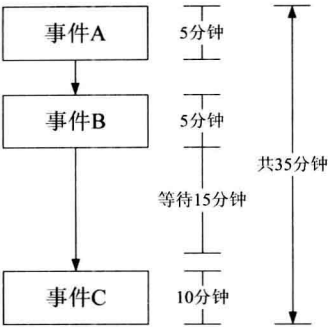


图 1-1 方法一

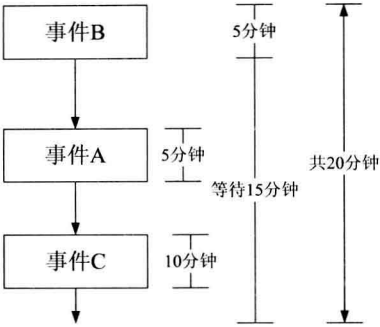


图 1-2 方法二

在上面的这个例子中，提到了两种方法，也就可以看做两种算法。第一种算法效率低，第二种算法效率高，但都达到了做完事情的目的。从这个例子可以看出，算法也是有好坏区别的，好的算法可以提供工作和生活的效率。算法的基本任务就是对于一个具体的问题，找到一个高效的处理方法，从而获得最佳的结果。

一个典型的算法一般都可以从其中抽象出 5 个特征：有穷性、确切性、输入、输出和可行性。下面结合上面的例子来分析这 5 个特征。

● 有穷性：

算法的指令或者步骤的执行次数是有限的，执行时间也是有限的。例如在上面的例子中，通过短短的几步就可以完成任务，而且执行时间都是有限的。

● 确切性：

算法的每一个指令或者步骤都必须有明确的定义和描述。例如在上面的例子中，为了完成做完三件事情的任务，每一步做什么事情都有明确的规定。

● 输入：

一个算法应该有相应的输入条件，用来刻画运算对象的初始情况。例如在上面的例子中，有三个待完成的事件（事件 A、事件 B 和事件 C）便是输入。

● 输出：

一个算法应该有明确的结果输出。这是容易理解的，没有得到结果的算法是毫无意义的。例如在上面的例子中，输出结果便是三件事情全部做完了。

● 可行性：

算法的执行步骤必须是可行的，且可以在有限时间内完成。例如在上面的例子中，每一个步骤都切实可行。其实，无法执行的步骤也是毫无意义的，解决不了任何实际问题。

目前，算法的应用非常广泛，常用的算法包括递推、递归、穷举、贪婪、分治、动态规划和迭代等多种。本书将逐步向读者展示各种算法的原理和应用。

1.2 算法的发展历史

关于算法的起源，可以追溯到中国古代公元前 1 世纪的《周髀算经》，它是算经的十书之一，原

名《周髀》，主要阐述古代中国的盖天说和四分历法。在唐朝的时候，被定为国子监明算科的教材之一，并改名为《周髀算经》。算法在古中国称为“演算法”。《周髀算经》中记载了勾股定理、开平方问题、等差级数问题等，里面用到了相当复杂的分数算法和开平方算法等。在随后的发展中，出现了割圆术、秦九韶算法和剩余定理等一些经典算法。

在西方，公元9世纪波斯数学家 al-Khwarizmi 提出了算法的概念。“算法”最初写为“algorism”，意思是采用阿拉伯数字的运算法则。到了18世纪，算法正式命名为现在的“algorithm”。由于汉字计算的不方便，导致中国古代算法的发展比较缓慢，而采用阿拉伯数字的西方国家则在随后的时间中发展迅速。例如著名的欧几里德算法（又称辗转相除法）就是典型的算法。

在历史上，大多数人都认可 Ada Byron 为第一个程序员。她在1842年编写巴贝奇分析机上的伯努利方程的求解算法程序，虽然未能执行，但奠定了计算机算法程序设计的基础。

后来，随着计算机的发展，在计算机上实现各种算法已成为可能。算法在计算机程序设计领域又得到了再次的重要发展。目前，几乎所有的程序员无论采用何种编程语言，都需要与算法打交道。

1.3 算法的分类

算法是一门古老且庞大的学科，随着历史的发展，演化出多种多样的算法。按照不同的应用和特性，可以分为不同的类别。

1. 按照应用来分类

按照算法的应用领域，也就是解决的问题，算法可以分为基本算法、数据结构相关的算法、几何算法、图论算法、规划算法、数值分析算法、加密/解密算法、排序算法、查找算法、并行算法和数论算法等。

2. 按照确定性来分类

按照算法结果的确定性来分类，可以分为确定性算法和非确定性算法。

- 确定性算法：这类算法在有限的时间内完成计算，得到的结果是唯一的，且经常取决于输入值。
- 非确定性算法：这类算法在有限的时间内完成计算，但是得到的结果往往不是唯一的，也就是存在多值性。

3. 按照算法的思路来分类

按照算法的思路来分类，算法可以分为递推算法、递归算法、穷举算法、贪婪算法、分治算法、动态规划算法和迭代算法等多种。

1.4 算法相关概念的区别

算法其实是一个很抽象的概念，往往需要依托于具体的实现方法才能体现其价值，例如在计算机编程中的算法、数值计算中的算法等。本书所重点讲解的便是算法在计算机编程中的应用。正是由于算法的抽象性，导致读者很容易产生混淆，这里有必要说明一些基本的概念。

1.4.1 算法与公式的关系

从前面所谈到的算法，让我们很容易联想到数学中的公式。公式也是解决某类问题，有特定的输入和结果输出，能在有限时间内完成，并且公式都是完全可以操作计算的。其实公式确实是提供了一种算法，但算法绝不完全等于公式。

公式是一种高度精简的计算方法，可以认为就是一种算法，其是人类智慧的结晶。而算法并不一定是公式，算法的形式可以比公式更复杂，解决的问题更加广泛。

1.4.2 算法与程序的关系

正如前面所述，算法是依托于具体的实现方式的。虽然我们一提到算法，就联想到计算机程序设计，但算法并非仅用于此。例如，在传统的笔算中，通过纸和笔按照一定的步骤完成的计算也是算法的应用；在速记中，人们通过特殊的方法来达到快速牢固记忆的目的，这也是一种算法的应用。

在计算机程序设计中，算法的体现更广泛，几乎每个程序都需要用到算法，只不过有些算法比较简单，有些算法比较复杂而已。

算法和程序设计语言是不同的。目前比较流行的程序设计语言，包括 Visual Basic、C、C++、C#、Java、Pascal、Delphi、PB 等。程序设计语言是算法实现的一种形式，也就是一种工具。我们往往需要首先熟悉程序设计语言的语法格式，然后才能使用这种程序语言编写合适的算法实现程序。学习一门程序设计语言是比较容易的，难的是如何正确合理地运用算法来编写求解问题。

1.4.3 算法与数据结构的关系

数据结构是数据的组织形式，可以用来表征特定的对象数据。在计算机程序设计中，操作的对象是各式各样的数据，这些数据往往拥有不同的数据结构，例如数组、结构体、联合、指针和链表等。因为不同的数据结构所采用的处理方法不同，计算的复杂程度也不同，因此算法往往是依赖于某种数据结构的。也就是说，数据结构是算法实现的基础。

计算机科学家尼克劳斯·沃思（Nikiklaus Wirth）曾提出了一个著名的公式：数据结构+算法=程序。后来，他将其出版为著名的书《数据结构+算法=程序》。我们从中可以看到算法和数据结构的关系。

有了前面的介绍，我们现在对程序、算法、数据结构、程序设计语言有了比较深刻的认识。如果给出一个公式的话，可以表述成如下形式：

$$\text{数据结构} + \text{算法} + \text{程序设计语言} = \text{程序}$$

这里，数据结构往往表示的是处理的对象，算法是计算和处理的核心方法，程序设计语言是算法的实现方法。这几者的综合便构成一个实实在在的程序。

注意：算法是解决问题的一个抽象方法和步骤，同一个算法在不同的语言中具有不同的实现形式，这依赖于数据结构的形式和程序设计语言的语法格式。

1.5 算法的表示

算法是为了解决实际问题的，问题简单，算法也简单；问题复杂，算法也一般相应地复杂。为

了便于交流和进行算法处理, 往往需要将算法进行描述, 也就是算法的表示。一般来说, 算法可以采用自然语言表示、流程图表示、N-S 图表示和伪代码表示。

1.5.1 自然语言表示

所谓自然语言, 就是自然地随文化演化的语言, 例如英语、汉语等。通俗地讲, 自然语言就是我们平时口头描述的语言。对于一些简单的算法, 可以采用自然语言来口头描述算法的执行过程, 例如前面的统筹安排事件的例子。

但是, 随着需求的发展, 很多算法都比较复杂, 很难用自然语言描述, 同时自然语言的表述烦琐难懂, 不利于发展和交流。因此, 需要采用其他的方法来进行表示。

其实, 我国古代早期的算法也可以看做是自然语言表示。正是由于这种复杂、烦琐的自然语言表示, 大大阻碍了中国古代算法的发展。这也正是为什么我国古代算法起源早, 但后期落后于西方国家的原因。

1.5.2 流程图表示

流程图是一种图形表示算法流程的方法, 其由一些图框和流程线组成, 如图 1-3 所示。其中, 图框表示各种操作的类型, 图框中的说明文字和符号表示该操作的内容, 流程线表示操作的先后次序。

流程图最大的优点便是简单直观、便于理解, 在计算机算法领域有着广泛的应用。例如, 计算两个输入数据 a 和 b 的最大值, 可以采用图 1-4 所示的流程图来表示。

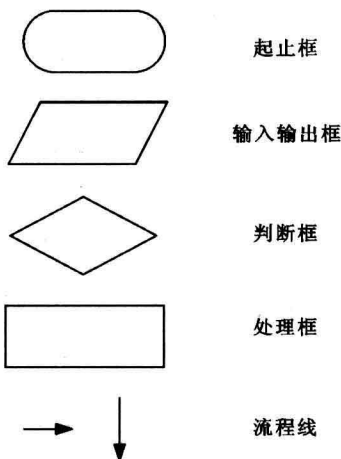


图 1-3 流程图的图元

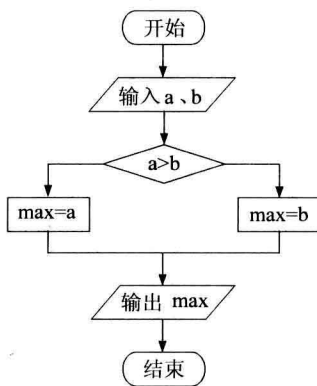


图 1-4 求最大值的流程图

在实际使用中, 一般采用如下三种流程结构。

1. 顺序结构

顺序结构是最简单的一种流程结构, 简单地一个接着一个地进行处理, 如图 1-5 所示。一般来说, 顺序结构适合于简单的算法。

2. 分支结构

分支结构常用于根据某个条件来决定算法的走向, 如图 1-6 所示。这里首先判断条件 P , 如果 P 成立, 则执行 B ; 否则执行 A , 然后再继续下面的算法。分支结构有时也称为条件结构。