

超过600分钟讲解视频和案例源代码倾囊相送
附赠5本电子书教程铺就Java程序员成长之路



Java

常用算法手册

【第3版】

宋娟◎编著

The Classic Algorithm Fundamental of Program Language Java

算法是一切程序设计的基础和灵魂，更是一位程序员编程水平高低的集中体现

- 涵盖广泛 —— 精炼的理论讲述嵌入经典算法示例，学习查询兼而有之
- 阐述到位 —— 算法思想、算法实现和经典面试题合理搭配，相辅相成
- 实例完善 —— 分析精准，注释精确，保证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

【第3版】

宋娟◎编著

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 14 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、游戏等领域中的应用；最后梳理和精选了一些经典的算法面试题，供读者开拓思维之用。

本书附赠光盘中包含第 14 章 PDF 文档，书中实例代码和一整套的 Java 学习教程，同时我们精心制作了与书中内容紧密结合的共计 20 讲的算法讲解视频，供读者学习之用。

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，也可为教师授课提供素材。

图书在版编目（CIP）数据

Java 常用算法手册/宋娟编著. —3 版. —北京：中国铁道出版社，2016.5
ISBN 978-7-113-17379-1

I. ①J… II. ①宋… III. ①JAVA 语言—程序设计—手册
IV. ①TP312-62

中国版本图书馆 CIP 数据核字(2016)第 076583 号

书 名：Java 常用算法手册（第 3 版）

作 者：宋娟 编著

责任编辑：荆 波

读者热线电话：010-63560056

责任印制：赵星辰

封面设计：MX DESIGN
STUDIO

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：北京尚品荣华印刷有限公司

版 次：2012 年 11 月第 1 版 2014 年 8 月第 2 版 2016 年 5 月第 3 版 2016 年 5 月第 1 次印刷

开 本：787mm×1092mm 1/16 印张：26 字数：670 千

书 号：ISBN 978-7-113-17379-1

定 价：59.80 元（附赠光盘）

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社读者服务部联系调换。电话：（010）51873174

打击盗版举报电话：（010）51873659

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而成。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，笔者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。

(3) 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。

(4) 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。

(5) 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局限于 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即可使用。

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 14 章内容。

第 1 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。本篇内容中既有对算法的深入诠释，更有作者对算法基本思想的经验分享。读者可通过本篇内容细致有序地建立起对算法理解的知识性框架。

第 2 篇 算法应用篇：本篇共 7 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。本篇可称为本书中的出彩部分，用实例嵌入知识讲解方式对各类算法进行了翔实地阐述；同时用一些贴近现实的生动实例对算法进行了有趣的表述，提升读者的编程能力和学习兴趣。

第 3 篇 算法面试篇：本篇共 4 章，详细讲解了常见面试中的算法。本篇中我们对面试题进行了细分，分别为上机类、逻辑推理类、教学能力类和基础-扩展算法面试类，通过

对这些面试类试题的详细分析，拓展读者的思维，提升能力。

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。

附书附赠光盘

本书光盘中包含两大部分，第一部分有：本书第 14 章的 PDF 电子档、书中源代码以及总计 20 讲的 Java 算法讲解视频，总时长约 600 分钟。第二部分则是一套电子书，以下是对这套电子书的详细说明。

算法是一个程序员成长的关键，但要完全掌握 Java 开发技能，还需要学习其他知识，本光盘是在纸质算法图书之外，额外提供一套“Java 程序员成长之路”，希望以一本算法书+一张配套电子书光盘，形成一个 Java 学习的完整解决方案。为此，我们设计了一个大学生能成长为 Java 程序员需要的全部图书的电子版，从基础到开发参考全包含：

- (1)《打通经脉——Java 基础入门编程详解》
- (2)《关键技术——JSP 与 JDBC 应用详解》
- (3)《独门架构——Java Web 开发应用详解》
- (4)《神兵利器——Eclipse 开发技术详解》
- (5)《开发参考——JSTL 网络标签与 SSH 类参考手册》

本系列电子书针对的是想成为 Java 专业开发人员的读者(大学高年级和工作前 3 年的职场新人)，给他们一揽子的解决方案，能够系统学习各个 Java 开发的知识和技能的方方面面。既能学会语言基础、面向对象这些基本内容，也能学到 Spring、Struts、Hibernate 这些真正 Java 开发人员必须掌握的高级开发框架技术方面的内容，目标是，学完全套图书的读者，能达到工作 1-2 年的 Java 程序员水平，覆盖大多数公司招聘 Java 程序员的技术要求范围。

本书由宁夏大学副教授宋娟编写，因时间仓促，不当之处，还请读者不吝指出，以期在以后的改版中进行改进。

编 者

2016 年 3 月

第 1 章 算法和实现算法的 Java 语法

1.1 建立算法初步概念.....	1
1.1.1 什么是算法.....	1
1.1.2 算法的发展历史.....	3
1.1.3 算法的分类.....	3
1.2 算法相关概念的区别.....	3
1.2.1 算法与公式的关系.....	4
1.2.2 算法与程序的关系.....	4
1.2.3 算法与数据结构的关系.....	4
1.3 算法的表示.....	4
1.3.1 自然语言表示.....	5
1.3.2 流程图表示.....	5
1.3.3 N-S 图表示.....	6
1.3.4 伪代码表示.....	7
1.4 算法的性能评价.....	7
1.4.1 时间复杂度.....	8
1.4.2 空间复杂度.....	8
1.5 一个算法实例.....	8
1.5.1 查找数字.....	8
1.5.2 创建项目.....	10
1.5.3 编译执行.....	11
1.6 Java 程序的基本结构.....	13
1.6.1 类是一个基本单元.....	13
1.6.2 main 方法.....	14
1.6.3 自定义方法.....	14
1.6.4 System.out.println 的使用.....	15
1.6.5 一个简单而完整的程序.....	15
1.7 顺序结构.....	16
1.8 分支结构.....	17
1.8.1 if...else 分支结构.....	17
1.8.2 if...else 嵌套.....	18
1.8.3 switch 语句.....	19
1.8.4 编程实例.....	20

1.9 循环结构.....	21
1.9.1 while 循环.....	22
1.9.2 do...while 循环.....	22
1.9.3 for 循环.....	23
1.9.4 编程实例.....	23
1.10 跳转结构.....	27
1.10.1 break.....	27
1.10.2 continue.....	27
1.10.3 编程实例.....	27
1.11 小结.....	29

第 2 章 数据结构

2.1 数据结构概述.....	30
2.1.1 什么是数据结构.....	30
2.1.2 数据结构中的基本概念.....	31
2.1.3 数据结构的内容.....	31
2.1.4 数据结构的分类.....	33
2.1.5 数据结构的几种存储方式.....	33
2.1.6 数据类型.....	34
2.1.7 常用的数据结构.....	35
2.1.8 选择合适的数据结构解决实际问题.....	36
2.2 线性表.....	36
2.2.1 什么是线性表.....	36
2.2.2 线性表的基本运算.....	37
2.3 顺序表结构.....	37
2.3.1 准备数据.....	38
2.3.2 初始化顺序表.....	38
2.3.3 计算顺序表长度.....	38
2.3.4 插入结点.....	39
2.3.5 追加结点.....	39
2.3.6 删除结点.....	40
2.3.7 查找结点.....	40
2.3.8 显示所有结点.....	41

第3章 基本算法思想

3.6.1 概率算法基本思想	115
3.6.2 概率算法实例	115
3.7 小结	117

第4章 排序算法

4.1 排序算法概述	118
4.2 冒泡排序算法	119
4.2.1 冒泡排序算法	119
4.2.2 冒泡排序算法实例	120
4.3 选择排序算法	122
4.3.1 选择排序算法	122
4.3.2 选择排序算法实例	123
4.4 插入排序算法	125
4.4.1 插入排序算法	125
4.4.2 插入排序算法实例	126
4.5 Shell 排序算法	127
4.5.1 Shell 排序算法	128
4.5.2 Shell 排序算法实例	129
4.6 快速排序算法	130
4.6.1 快速排序算法	131
4.6.2 快速排序算法实例	132
4.7 堆排序算法	134
4.7.1 堆排序算法	134
4.7.2 堆排序算法实例	138
4.8 合并排序算法	141
4.8.1 合并排序算法	141
4.8.2 合并排序算法实例	144
4.9 排序算法的效率	147
4.10 排序算法的其他应用	147
4.10.1 反序排序	147
4.10.2 字符串数组的排序	150
4.10.3 字符串的排序	153
4.11 小结	155

第5章 查找算法

5.1 查找算法概述	156
5.2 顺序查找	157
5.2.1 顺序查找算法	157
5.2.2 顺序查找操作实例	157

5.3 折半查找	159
5.3.1 折半查找算法	159
5.3.2 折半查找操作实例	161
5.4 数据结构中的查找算法	163
5.4.1 顺序表结构中的查找算法	164
5.4.2 链表结构中的查找算法	167
5.4.3 树结构中的查找算法	171
5.4.4 图结构中的查找算法	171
5.5 小结	172

第6章 基本数学问题

6.1 判断闰年	173
6.2 多项式计算	175
6.2.1 一维多项式求值	175
6.2.2 二维多项式求值	177
6.2.3 多项式乘法	179
6.2.4 多项式除法	181
6.3 随机数生成算法	183
6.3.1 Java 语言中的随机方法	183
6.3.2 [0, 1]之间均匀分布的随机数 算法	185
6.3.3 产生任意范围的随机数	187
6.3.4 [m, n]之间均匀分布的随机 整数算法	188
6.3.5 正态分布的随机数生成算法	189
6.4 复数运算	191
6.4.1 简单的复数运算	191
6.4.2 复数的幂运算	194
6.4.3 复指数运算	195
6.4.4 复对数运算	196
6.4.5 复正弦运算	198
6.4.6 复余弦运算	199
6.5 阶乘	200
6.5.1 使用循环来计算阶乘	200
6.5.2 使用递归来计算阶乘	201
6.6 计算 π 的近似值	202
6.6.1 割圆术	203
6.6.2 蒙特卡罗算法	206
6.6.3 级数公式	208

6.7	矩阵运算.....	210
6.7.1	矩阵加法.....	210
6.7.2	矩阵减法.....	211
6.7.3	矩阵乘法.....	213
6.8	方程求解.....	215
6.8.1	线性方程求解—— 高斯消元法.....	215
6.8.2	非线性方程求解—— 二分法.....	220
6.8.3	非线性方程求解—— 牛顿迭代法.....	222
6.9	小结.....	225

第 7 章 数据结构问题

7.1	动态数组排序.....	226
7.1.1	动态数组的存储和排序.....	226
7.1.2	动态数组排序实例.....	227
7.2	约瑟夫环.....	230
7.2.1	简单约瑟夫环算法.....	230
7.2.2	简单约瑟夫环求解.....	232
7.2.3	复杂约瑟夫环算法.....	233
7.2.4	复杂约瑟夫环求解.....	234
7.3	城市之间的最短总距离.....	237
7.3.1	最短总距离算法.....	237
7.3.2	最短总距离求解.....	240
7.4	最短路径.....	244
7.4.1	最短路径算法.....	245
7.4.2	最短路径求解.....	247
7.5	括号匹配.....	252
7.5.1	括号匹配算法.....	252
7.5.2	括号匹配求解.....	254
7.6	小结.....	257

第 8 章 数论问题

8.1	数论概述.....	258
8.1.1	数论概述.....	258
8.1.2	数论的分类.....	259
8.1.3	初等数论.....	260
8.1.4	本章用到的基本概念.....	260

8.2	完全数.....	261
8.2.1	什么是完全数.....	261
8.2.2	计算完全数算法.....	262
8.3	亲密数.....	264
8.3.1	什么是亲密数.....	264
8.3.2	计算亲密数算法.....	264
8.4	水仙花数.....	267
8.4.1	什么是水仙花数.....	267
8.4.2	计算水仙花数算法.....	268
8.5	自守数.....	270
8.5.1	什么是自守数.....	270
8.5.2	计算自守数算法.....	271
8.6	最大公约数.....	274
8.6.1	计算最大公约数算法—— 辗转相除法.....	275
8.6.2	计算最大公约数算法—— Stein 算法.....	275
8.6.3	计算最大公约数示例.....	277
8.7	最小公倍数.....	278
8.8	素数.....	280
8.8.1	什么是素数.....	280
8.8.2	计算素数算法.....	280
8.9	回文素数.....	282
8.9.1	什么是回文素数.....	282
8.9.2	计算回文素数算法.....	282
8.10	平方回文数.....	285
8.10.1	什么是平方回文数.....	285
8.10.2	计算平方回文数算法.....	285
8.11	分解质因数.....	287
8.12	小结.....	289

第 9 章 算法经典趣题

9.1	百钱买百鸡.....	290
9.1.1	百钱买百鸡算法.....	290
9.1.2	百钱买百鸡求解.....	291
9.2	五家共井.....	292
9.2.1	五家共井算法.....	292
9.2.2	五家共井求解.....	293
9.3	鸡兔同笼.....	295

9.3.1	鸡兔同笼算法	295
9.3.2	鸡兔同笼求解	295
9.4	猴子吃桃	296
9.4.1	猴子吃桃算法	296
9.4.2	猴子吃桃求解	297
9.5	舍罕王赏麦	298
9.5.1	舍罕王赏麦问题	298
9.5.2	舍罕王赏麦求解	299
9.6	汉诺塔	300
9.6.1	汉诺塔算法	301
9.6.2	汉诺塔求解	302
9.7	窃贼问题	303
9.7.1	窃贼问题算法	303
9.7.2	窃贼问题求解	305
9.8	马踏棋盘	308
9.8.1	马踏棋盘算法	308
9.8.2	马踏棋盘求解	310
9.9	八皇后问题	312
9.9.1	八皇后问题算法	312
9.9.2	八皇后问题求解	314
9.10	寻找假银币	315
9.10.1	寻找假银币算法	316
9.10.2	寻找假银币求解	318
9.11	青蛙过河	320
9.11.1	青蛙过河算法	320
9.11.2	青蛙过河求解	321
9.12	三色旗	325
9.12.1	三色旗算法	325
9.12.2	三色旗求解	326
9.13	渔夫捕鱼	328
9.13.1	渔夫捕鱼算法	328
9.13.2	渔夫捕鱼求解	329
9.14	爱因斯坦的阶梯	330
9.14.1	爱因斯坦的阶梯算法	330
9.14.2	爱因斯坦的阶梯求解	331
9.15	兔子产仔	332
9.15.1	兔子产仔算法	332
9.15.2	兔子产仔求解	333
9.16	常胜将军	334

9.16.1	常胜将军算法	334
9.16.2	常胜将军求解	335
9.17	新郎和新娘	336
9.17.1	新郎和新娘算法	337
9.17.2	新郎和新娘求解	337
9.18	三色球	339
9.18.1	三色球算法	339
9.18.2	三色球求解	340
9.19	小结	341

第 10 章 游戏中的算法

10.1	洗扑克牌算法	342
10.1.1	洗扑克牌算法	342
10.1.2	洗扑克牌实例	343
10.2	取火柴游戏算法	346
10.2.1	取火柴游戏算法	346
10.2.2	取火柴游戏实例	347
10.3	十点半算法	349
10.3.1	十点半算法	349
10.3.2	十点半游戏实例	355
10.4	生命游戏	361
10.4.1	生命游戏的原理	362
10.4.2	生命游戏的算法	363
10.4.3	生命游戏实例	364
10.5	小结	369

第 11 章 简单 Java 上机面试题

11.1	打印九九乘法口诀表	370
11.2	获得任意一个时间的下一天的时间	371
11.3	将某个时间以固定格式转化成字符串	373
11.4	怎样截取字符串	374
11.5	怎样实现元素互换	375
11.6	怎样实现元素排序	377
11.7	怎样实现 Singleton 模式编程	379
11.8	怎样实现金额转换	380
11.9	如何判断回文数字	383

11.10	小结.....	384
-------	---------	-----

第 12 章 逻辑推理类面试题

12.1	脑筋急转弯.....	385
12.1.1	中国有多少辆汽车.....	385
12.1.2	下水道的盖子为什么是圆形的.....	386
12.1.3	分蛋糕.....	387
12.2	逻辑推理.....	387
12.2.1	哪个开关控制哪盏灯.....	388
12.2.2	戴帽子.....	388
12.2.3	海盗分金.....	389
12.2.4	罪犯认罪.....	390
12.2.5	找出质量不相同的球.....	391
12.2.6	有多少人及格.....	391
12.2.7	他说的是真话吗.....	392
12.3	计算推理.....	393
12.3.1	倒水问题.....	393
12.3.2	骗子购物.....	394
12.3.3	求最大的连续组合值（华为校园招聘笔试题）.....	395
12.3.4	洗扑克牌（乱数排列）.....	396
12.3.5	字符移动（金山笔试题）.....	398
12.4	小结.....	399

第 13 章 数学能力测试

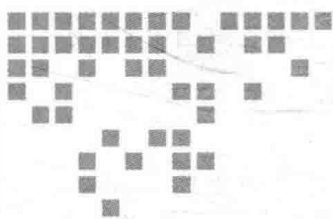
13.1	100 盏灯.....	400
------	-------------	-----

13.2	用一笔画出经过 9 个点的 4 条直线.....	401
13.3	时针、分针和秒针重合问题.....	402
13.4	怎样拿到第 100 号球.....	405
13.5	烧绳计时.....	406

（注：以下内容读者可在附赠光盘中阅读学习。）

第 14 章 算法面试题

14.1	基础算法.....	408
14.1.1	字符串匹配.....	408
14.1.2	哥德巴赫猜想的近似证明... ..	411
14.1.3	将一个正整数分解质因数... ..	413
14.1.4	怎样实现金额转换.....	415
14.1.5	数字排列.....	419
14.1.6	数字拆解.....	421
14.1.7	数字组合.....	423
14.2	思维扩展算法.....	426
14.2.1	蛇形打印.....	426
14.2.2	24 点算法.....	428
14.2.3	双色球随机摇号.....	432
14.2.4	巧妙过桥.....	435
14.2.5	猴子吃桃.....	439
14.2.6	天平称物.....	439
14.2.7	掷骰子游戏.....	441
14.3	小结.....	444



Chapter 1 第 1 章

算法和实现算法的 Java 语法

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、本书具体实现算法的 Java 语言语法和应用等。

1.1 建立算法初步概念

很多开发者都知道“程序=数据结构+算法”这个著名的公式，但却并不真正明白算法的定义或者概念。

1.1.1 什么是算法

究竟什么是算法（algorithm）呢？从字面意义上理解，算法即用于计算的方法，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中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。

举一个例子来分析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，如烧水等待水开的过程；
- 做事件 C 需要耗费 10 分钟。

那么应该怎样来做这三件事情呢？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，总的耗时为 $5 + (5 + 15) + 10 = 35$ 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，总的耗时为 $5 + 15 = 20$ 分钟，效率明显提高，如图 1-2 所示。

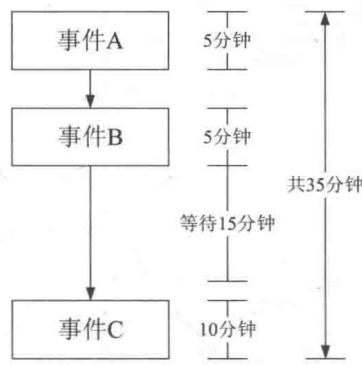


图 1-1 方法一

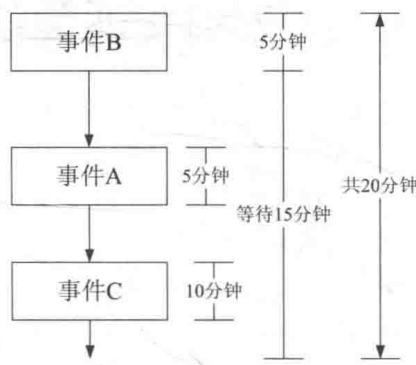


图 1-2 方法二

在上述例子中提到的两种方法就可以看作两种算法。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，好的算法可以提高工作的效率。算法的基本任务是针对一个具体的问题，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。下面结合上述例子来分析这 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），这三个事件便是输入。
- 输出：一个算法应该有明确的结果输出。这是容易理解的，因为没有得到结果的算法毫无意义。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。本书将逐步向读者展示各种算法的原理和应用。

1.1.2 算法的发展历史

关于算法的起源，可以追溯到我国古代公元前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，此书被定为国子监明算科的教材之一，并改名为《周髀算经》。算法在我国古代称为“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，其中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 algorithm。由于汉字计算的不方便，导致我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，Ada Byron 被认为是第一个程序员。她在1842年编写的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，在计算机中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重要发展。目前，几乎所有的程序员编程时，无论采用何种编程语言，都需要与算法打交道。

1.1.3 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，算法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，即解决的问题，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，算法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，即存在多值性。

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算法。

1.2 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，如在计算机编程中的算法、数值计算中的算法等。本书重点讲解的便是算法在计算机编程中的应用。算法具有抽象性，容易混淆，这里有必要说明一些基本的概念。

1.2.1 算法与公式的关系

根据前面所谈到的算法,读者很容易联想到数学中的公式。公式用于解决某类问题,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。公式确实提供了一种算法,但算法绝不等于公式。

公式是一种高度精简的计算方法,可以认为就是一种算法,它是人类智慧的结晶。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。

1.2.2 算法与程序的关系

正如前面所述,算法依托于具体的实现方式。虽然一提到算法,就使人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,也是一种工具。读者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。

1.2.3 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,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,因此算法往往依赖于某种数据结构。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(Niklaus Wirth)曾提出了一个著名的公式:数据结构+算法=程序。后来,他为此著有《数据结构+算法=程序》一书。从中可以看到算法和数据结构的关系。

学习了前面的介绍,读者应对程序、算法、数据结构、程序设计语言有一个比较深刻的认识。如果给出一个公式,则可以表述成如下形式:

数据结构+算法+程序设计语言=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。

1.3 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,算法也相应复杂。为

了便于交流和进行算法处理，往往需要将算法进行描述，即算法的表示。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。

1.3.1 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，如英语、汉语等。通俗地讲，自然语言就是人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，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作自然语言表示。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。这也正是为何我国古代算法起源早，但后来落后于西方国家的原因。

1.3.2 流程图表示

流程图是用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。

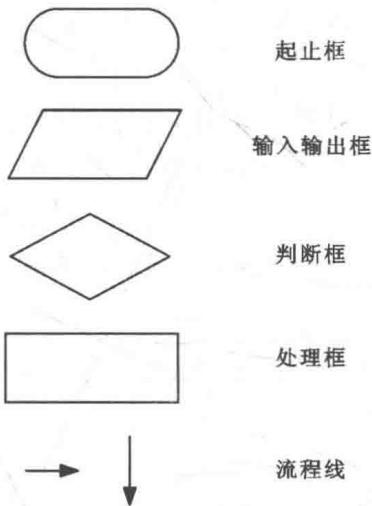


图 1-3 流程图的图元

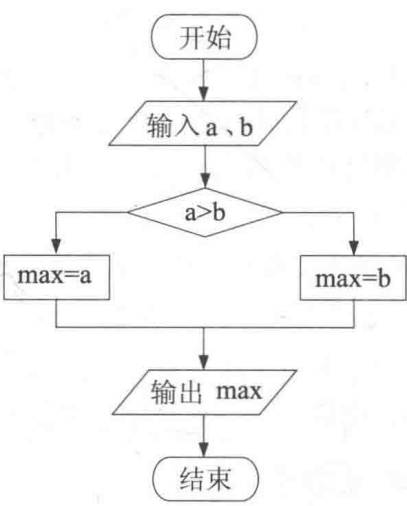


图 1-4 求最大值的流程图

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。

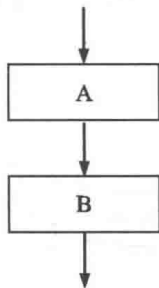


图 1-5 顺序结构

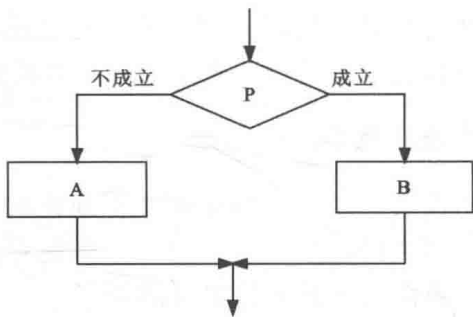


图 1-6 分支结构

3) 循环结构

循环结构常用于需要反复执行的算法操作，按照循环的方式，可以分为当型循环结构和直到型循环结构，分别如图 1-7 和图 1-8 所示。

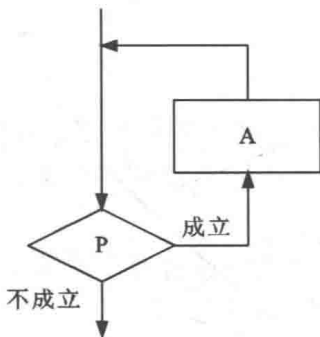


图 1-7 当型循环结构

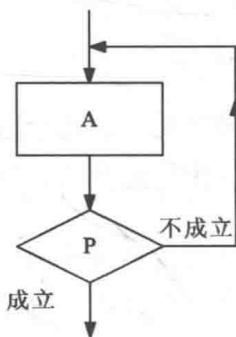


图 1-8 直到型循环结构

当型循环结构和直到型循环结构的区别如下：

- 当型循环结构先对条件进行判断，然后再执行，一般采用 while 语句来实现；
- 直到型循环结构先执行，然后再对条件进行判断，一般采用 until、do...while 语句来实现。

注意：无论当型循环结构还是直到型循环结构，都需要进行合适的处理，以保证能够跳出循环；否则构成死循环是没有任何意义的。

一般来说，采用上述三种流程结构，可以完成所有的算法任务。通过合理安排流程结构，可以构成结构化的程序，这样便于算法程序的开发和交流。

1.3.3 N-S 图表示

N-S 图也称为盒图或者 CHAPIN 图，1973 年由美国学者 I.Nassi 和 B.Shneiderman 提出。他们发现采用流程图可以清楚地表示算法或程序的运行过程，但其中的流程线并不是必需的，因此而创立了 N-S 图。在 N-S 图中，将整个程序写在一个大框图内，这个大框图由若干个小的基本框图构成。采用 N-S 图也可以方便地表示流程图的内容。

采用 N-S 图表示的顺序结构，如图 1-9 所示。采用 N-S 图表示的分支结构，如图 1-10 所示。采用 N-S 图表示的当型循环结构，如图 1-11 所示。采用 N-S 图表示的直到型循环结构，如图 1-12 所示。