

提供专用邮箱和专属学习论坛，让本书成为读者学习算法的阶梯。

C/C++

常用算法手册

The Classic Algorithm Fundamental of
Program Language C/C++

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**。

● 秦姣华 向旭宇 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。

C/C++

常用算法手册

The Classic Algorithm Fundamental of
Program Language C/C++

● 秦姣华 向旭宇 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

计算机技术的发展和普及不仅改变了人们的生活和娱乐方式,也改变了人们的工作方式,这其中最为重要的便是计算机编程技术。现代的设计任务大多通过代码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。

本书分3篇,共13章,“第1篇 算法基础篇”介绍了算法概述,重点分析了数据结构和基本算法思想;“第2篇 算法基本应用篇”详细讲解了算法在排序、查找、数值计算、数论、经典趣题和游戏中的应用;“第3篇 算法高级应用篇”讲解了算法的一些高级应用技术,包括在密码学和数据压缩/解压缩中的应用。

本书知识点覆盖全面、结构安排紧凑、讲解详细、示例丰富。全书对每一个知识点都给出了相应的算法及应用示例。虽然这些例子都是以C语言来编写的,但是算法并不局限于C语言。如果读者采用其他编程语言,例如C++、C#、VB、Java等,根据其语法格式进行适当的修改即可。

本书主要定位于有一定C/C++语言编程基础、想通过学习算法与数据结构提升编程水平的读者,也可作为具有一定编程经验的程序员以及大中专院校学生学习数据结构和算法的参考书。

图书在版编目(CIP)数据

C/C++常用算法手册 / 秦姣华, 向旭宇编著. --北京: 中国铁道出版社, 2011.5

ISBN 978-7-113-12570-7

I. ①C… II. ①秦… ②向… III. ①C语言—程序设计—手册 IV. ①TP312-62

中国版本图书馆CIP数据核字(2011)第020793号

书 名: C/C++常用算法手册
作 者: 秦姣华 向旭宇 编著

策划编辑: 荆 波
责任编辑: 荆 波
特邀编辑: 王 惠
封面设计: 付 巍
责任印制: 李 佳

读者服务热线: 400-668-0820

编辑助理: 王 宏
封面制作: 郑少云

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)

印 刷: 北京铭成印刷有限公司

版 次: 2011年5月第1版 2011年5月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 26.25 字数: 617千

印 数: 3 500册

书 号: ISBN 978-7-113-12570-7

定 价: 49.00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。

计算机程序设计是信息化进程中最为重要的设计手段。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。

但是，数据结构和算法理论性很强，读者在学习的过程中会感到枯燥和吃力，往往学习一段时间后便丧失了兴趣，这就使得学习的效率大大降低。如何才能提高读者学习的兴趣，使读者能快速掌握数据结构和算法的知识呢？其实读者需要的不仅是理论知识，还需要了解这些知识点的代码实现以及应用示例。另外，对知识背景的介绍和理解往往能激发读者学习的兴趣。编者便从这些基本点出发，为读者编写了一本可以轻松学习数据结构和算法的参考书。

C语言加上它的衍生语言C++是目前最为流行的编程语言。本书中所有的算法及示例都是采用C语言进行编写的，当然因为基本语法一样，所以同时也能在C++环境下运行。但是，这些算法并不局限于C语言。如果读者采用其他编程语言，例如C#、VB、Java等，根据其语法格式进行适当的修改便可使用。毕竟算法是核心，C和C++语言也是众多其他语言发展时的重要参考，很多语法特点也相同。

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，以保证其质量和延长其生命力。与其他书籍相比，本书有如下特色：

(1) 本书由浅入深、循序渐进地带领读者学习数据结构和算法的知识。

(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和算法实现，同时还给出了完整的应用示例，每个示例都可以通过编译执行，使读者可以快速掌握相应的知识点在程序设计中的应用。

(4) 本书在介绍各个知识点的同时，尽量结合历史背景并给出问题的完整分析，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，增加了图书的易读性。

(5) 本书对每一个示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，使读者在学习时更加容易理解。

本书的内容

本书以实用性、系统性、完整性和前沿性为特点，详细介绍了算法的基本思想和不同领域的应用示例。本书内容共分3篇13章。

第1篇是算法基础篇，共分为3章，详细介绍了算法和数据结构的相关知识。

第1章为算法概述，介绍了什么是算法，以及算法的发展、分类、基本概念、表示以及性能评价等。同时，还通过一个示例分析算法的编写和编译执行等。最后，本章还介绍了一些算法的最新进展，扩展读者的知识面。

第2章为数据结构，首先介绍了数据结构基本概念，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2篇为算法基本应用篇，共分为8章，详细讲解了算法在排序、查找、数值计算、数论、经典趣题和游戏中的应用。

第4章为排序算法，详细介绍冒泡、选择、插入、Shell、快速、堆排等各种排序算法及其应用示例。

第5章为查找算法，详细讲解了顺序、折半等各种查找算法及其应用示例。

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，并通过完整的示例来演示算法的应用。

第7章为复杂的数值计算算法，介绍了在科学和工程计算中常用的插值、数值积分、开平方、求极值以及一些特殊函数的数值计算算法。

第8章为经典数据结构问题，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内容。

第9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一些基本数据的计算算法。

第10章为算法经典趣题，介绍了一些历史上非常有名的算法问题和智力趣题等。这些经典趣题是历史累积的算法财富，通过这些有趣的问题，可以提升读者的编程能力和激发学习算法的兴趣。

第11章为游戏中的算法，讲解了一些经典游戏中的算法，包括洗扑克牌算法、取火柴游戏算法、10点半算法以及生命游戏算法。

第3篇为算法高级应用篇，共分为2章，详细讲解了算法在密码学和数据压缩/解压缩中的应用。

第 12 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 13 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，最后结合示例讲解了压缩/解压缩算法的应用。

适合的读者

- 系统开发人员
- 程序设计初学者
- C 程序员
- 电脑培训班学员
- 计算机爱好者
- 大学、大专等相关专业的学生及教师

本书结构紧凑，涉及的知识点也较全面，内容翔实，示例丰富。由于本书内容较多，编写时间仓促，书中如有疏漏或不足之处，恳请广大读者提出宝贵意见，以便进一步改进。

增值服务

一本真正的好书，从策划到出版上市会凝聚很多人行之有效的想法以及智慧，它不但为读者打开一扇学习知识的门，更要为读者在书本之外搭建起一条提升能力的阶梯。为了让本书更加完善，读者在学习本书的过程中如果发现有不明白的地方或者有更好的算法和其他建议，我们真诚地欢迎您发邮件到 1057762679@qq.com 和我们交流，相互学习和一起提升。另外，阅读完本书后，读者如果有增加学习深度和了解其他算法相关技术的想法，也可到本书专有的读者服务网站 <http://www.rzchina.net/forum/55> 上交流学习心得、阅读相关技术，还可以阅读在实际工作中可能会用到的其他算法技术资料，该资料内容至少有 600 页。同时，我们还会在这个网站上提供辅助视频教学服务，供读者下载或者在线学习。

编 者

2011 年 1 月

第 1 篇 算法基础篇

第 1 章 算法概述

1.1 什么是算法	2
1.2 算法的发展历史	3
1.3 算法的分类	4
1.4 算法相关概念的区别	4
1.5 算法的表示	5
1.5.1 自然语言表示	5
1.5.2 流程图表示	6
1.5.3 N-S 图表示	7
1.5.4 伪代码表示	7
1.6 算法的性能评价	8
1.7 算法实例	9
1.7.1 查找数字	9
1.7.2 创建项目	11
1.7.3 编译执行	12
1.8 算法的新进展	13
1.9 小结	14

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
2.1.6 数据类型	19
2.1.7 常用的数据结构	20
2.1.8 选择合适的数据结构解决实际 问题	21

2.2 线性表	21
2.2.1 什么是线性表	21
2.2.2 线性表的基本运算	22
2.3 顺序表结构	23
2.3.1 准备数据	23
2.3.2 初始化顺序表	24
2.3.3 计算顺序表长度	24
2.3.4 插入结点	24
2.3.5 追加结点	25
2.3.6 删除结点	25
2.3.7 查找结点	25
2.3.8 显示所有结点	26
2.3.9 顺序表操作示例	26
2.4 链表结构	30
2.4.1 什么是链表结构	30
2.4.2 准备数据	31
2.4.3 追加结点	31
2.4.4 插入头结点	33
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	36
2.4.9 显示所有结点	36
2.4.10 链表操作示例	37
2.5 栈结构	41
2.5.1 什么是栈结构	41
2.5.2 准备数据	42
2.5.3 初始化栈结构	42
2.5.4 判断空栈	43
2.5.5 判断满栈	43
2.5.6 清空栈	43

2.5.7	释放空间.....	44
2.5.8	入栈.....	44
2.5.9	出栈.....	44
2.5.10	读结点数据.....	45
2.5.11	栈结构操作示例.....	45
2.6	队列结构.....	48
2.6.1	什么是队列结构.....	48
2.6.2	准备数据.....	49
2.6.3	初始化队列结构.....	49
2.6.4	判断空队列.....	50
2.6.5	判断满队列.....	50
2.6.6	清空队列.....	50
2.6.7	释放空间.....	51
2.6.8	入队列.....	51
2.6.9	出队列.....	51
2.6.10	读结点数据.....	52
2.6.11	计算队列长度.....	52
2.6.12	队列结构操作示例.....	53
2.7	树结构.....	56
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.....	61
2.7.5	初始化二叉树.....	61
2.7.6	添加结点.....	62
2.7.7	查找结点.....	63
2.7.8	获取左子树.....	64
2.7.9	获取右子树.....	64
2.7.10	判断空树.....	65
2.7.11	计算二叉树深度.....	65
2.7.12	清空二叉树.....	65
2.7.13	显示结点数据.....	66

2.7.14	遍历二叉树.....	66
2.7.15	树结构操作示例.....	68
2.8	图结构.....	71
2.8.1	什么是图结构.....	71
2.8.2	图的基本概念.....	72
2.8.3	准备数据.....	76
2.8.4	创建图.....	78
2.8.5	清空图.....	79
2.8.6	显示图.....	79
2.8.7	遍历图.....	80
2.8.8	图结构操作示例.....	81
2.9	小结.....	84

第3章 基本算法思想

3.1	常用算法思想概述.....	85
3.2	穷举算法思想.....	85
3.2.1	穷举算法基本思想.....	86
3.2.2	穷举算法示例.....	86
3.3	递推算法思想.....	88
3.3.1	递推算法基本思想.....	88
3.3.2	递推算法示例.....	88
3.4	递归算法思想.....	90
3.4.1	递归算法基本思想.....	90
3.4.2	递归算法示例.....	90
3.5	分治算法思想.....	92
3.5.1	分治算法基本思想.....	92
3.5.2	分治算法示例.....	92
3.6	概率算法思想.....	96
3.6.1	概率算法基本思想.....	96
3.6.2	概率算法示例.....	97
3.7	小结.....	98

第2篇 算法基本应用篇

第4章 排序算法

4.1	排序算法概述.....	100
4.2	冒泡排序法.....	101
4.2.1	冒泡排序算法.....	101

4.2.2	冒泡排序算法示例.....	102
4.3	选择排序法.....	104
4.3.1	选择排序算法.....	104
4.3.2	选择排序算法示例.....	105
4.4	插入排序法.....	107
4.4.1	插入排序算法.....	107

4.4.2	插入排序算法示例	108
4.5	Shell 排序法	110
4.5.1	Shell 排序算法	110
4.5.2	Shell 排序算法示例	111
4.6	快速排序法	113
4.6.1	快速排序算法	113
4.6.2	快速排序算法示例	114
4.7	堆排序法	116
4.7.1	堆排序算法	116
4.7.2	堆排序算法示例	121
4.8	合并排序法	123
4.8.1	合并排序算法	123
4.8.2	合并排序算法示例	126
4.9	排序算法的效率	129
4.10	排序算法的其他应用	130
4.10.1	反序排序	130
4.10.2	字符串数组的排序	132
4.10.3	字符串的排序	135
4.11	小结	137

第 5 章 查找算法

5.1	查找算法概述	138
5.2	顺序查找	138
5.2.1	顺序查找算法	139
5.2.2	顺序查找操作示例	139
5.3	折半查找	141
5.3.1	折半查找算法	141
5.3.2	折半查找操作示例	142
5.4	数据结构中的查找算法	145
5.4.1	顺序表结构中的查找算法	145
5.4.2	链表结构中的查找算法	148
5.4.3	树结构中的查找算法	151
5.4.4	图结构中的查找算法	152
5.5	小结	153

第 6 章 基本数学问题

6.1	判断闰年	154
6.2	多项式计算	156
6.2.1	一维多项式求值	156

6.2.2	二维多项式求值	158
6.2.3	多项式乘法	160
6.2.4	多项式除法	161
6.3	随机数生成算法	164
6.4	复数运算	171
6.4.1	简单的复数运算	172
6.4.2	复数的幂运算	174
6.4.3	复指数运算	176
6.4.4	复对数运算	177
6.4.5	复正弦运算	178
6.4.6	复余弦运算	179
6.5	阶乘	180
6.6	计算 π 的近似值	183
6.6.1	割圆术	183
6.6.2	蒙特卡罗算法	185
6.6.3	级数公式	187
6.7	矩阵运算	190
6.7.1	矩阵加法	190
6.7.2	矩阵减法	191
6.7.3	矩阵乘法	193
6.8	方程求解	195
6.8.1	线性方程求解——高斯 消元法	195
6.8.2	非线性方程求解—— 二分法	200
6.8.3	非线性方程求解——牛顿 迭代法	202
6.9	小结	205

第 7 章 复杂的数值计算算法

7.1	拉格朗日插值	206
7.1.1	拉格朗日插值算法	206
7.1.2	拉格朗日插值示例	207
7.2	数值积分	210
7.2.1	数值积分算法	210
7.2.2	数值积分示例	211
7.3	开平方	213
7.3.1	开平方算法	213
7.3.2	开平方示例	213

7.4	极值问题的求解算法.....	215
7.4.1	极值求解算法.....	215
7.4.2	极值求解示例.....	217
7.5	特殊函数的计算算法.....	221
7.5.1	伽玛函数.....	221
7.5.2	贝塔函数.....	224
7.5.3	正弦积分函数.....	228
7.5.4	余弦积分函数.....	231
7.5.5	指数积分函数.....	235
7.6	小结.....	239

第8章 经典数据结构问题

8.1	动态数组排序.....	240
8.1.1	动态数组的存储和排序.....	240
8.1.2	动态数组排序示例.....	241
8.2	约瑟夫环.....	243
8.2.1	简单约瑟夫环算法.....	243
8.2.2	简单约瑟夫环求解.....	245
8.2.3	复杂约瑟夫环算法.....	247
8.2.4	复杂约瑟夫环求解.....	248
8.3	城市之间的最短总距离.....	250
8.3.1	最短总距离算法.....	250
8.3.2	最短总距离求解.....	253
8.4	最短路径.....	257
8.4.1	最短路径算法.....	258
8.4.2	最短路径求解.....	260
8.5	括号匹配.....	265
8.5.1	括号匹配算法.....	265
8.5.2	括号匹配求解.....	267
8.6	小结.....	270

第9章 数论问题

9.1	数论.....	271
9.1.1	数论概述.....	271
9.1.2	数论的分类.....	272
9.1.3	初等数论.....	273
9.1.4	基本概念.....	273
9.2	完全数.....	274
9.2.1	完全数概述.....	274

9.2.2	计算完全数算法.....	275
9.3	亲密数.....	277
9.3.1	亲密数概述.....	277
9.3.2	计算亲密数算法.....	277
9.4	水仙花数.....	280
9.4.1	水仙花数概述.....	280
9.4.2	计算水仙花数算法.....	281
9.5	自守数.....	283
9.5.1	自守数概述.....	283
9.5.2	计算自守数算法.....	284
9.6	最大公约数.....	287
9.6.1	计算最大公约数算法——辗转 相除法.....	287
9.6.2	计算最大公约数算法——Stein 算法.....	288
9.6.3	计算最大公约数示例.....	289
9.7	最小公倍数.....	290
9.8	素数.....	292
9.8.1	素数概述.....	292
9.8.2	计算素数算法.....	292
9.9	回文素数.....	294
9.9.1	回文素数概述.....	294
9.9.2	计算回文素数算法.....	294
9.10	平方回文数.....	297
9.10.1	平方回文数概述.....	297
9.10.2	计算平方回文数算法.....	297
9.11	分解质因数.....	299
9.12	小结.....	301

第10章 算法经典趣题

10.1	百钱买百鸡.....	302
10.1.1	百钱买百鸡算法.....	302
10.1.2	百钱买百鸡求解.....	303
10.2	五家共井.....	304
10.2.1	五家共井算法.....	304
10.2.2	五家共井求解.....	305
10.3	鸡兔同笼.....	307
10.3.1	鸡兔同笼算法.....	307
10.3.2	鸡兔同笼求解.....	308

10.4 猴子吃桃	308	10.14.1 爱因斯坦的阶梯算法	341
10.4.1 猴子吃桃算法	308	10.14.2 爱因斯坦的阶梯求解	342
10.4.2 猴子吃桃求解	309	10.15 兔子产仔	342
10.5 舍罕王赏麦	310	10.15.1 兔子产仔算法	343
10.5.1 舍罕王赏麦问题	310	10.15.2 兔子产仔求解	343
10.5.2 舍罕王赏麦求解	311	10.16 常胜将军	344
10.6 汉诺塔	312	10.16.1 常胜将军算法	344
10.6.1 汉诺塔算法	312	10.16.2 常胜将军求解	345
10.6.2 汉诺塔求解	314	10.17 新郎和新娘	346
10.7 窃贼问题	315	10.17.1 新郎和新娘算法	347
10.7.1 窃贼问题算法	315	10.17.2 新郎和新娘求解	348
10.7.2 窃贼问题求解	317	10.18 三色球	349
10.8 马踏棋盘	320	10.18.1 三色球算法	349
10.8.1 马踏棋盘算法	320	10.18.2 三色球求解	350
10.8.2 马踏棋盘求解	321	10.19 小结	351
10.9 八皇后问题	323		
10.9.1 八皇后问题算法	324		
10.9.2 八皇后问题求解	325		
10.10 寻找假银币	327		
10.10.1 寻找假银币算法	327		
10.10.2 寻找假银币求解	329		
10.11 青蛙过河	331		
10.11.1 青蛙过河算法	331		
10.11.2 青蛙过河求解	333		
10.12 三色旗	335		
10.12.1 三色旗算法	335		
10.12.2 三色旗求解	337		
10.13 渔夫捕鱼	339		
10.13.1 渔夫捕鱼算法	339		
10.13.2 渔夫捕鱼求解	340		
10.14 爱因斯坦的阶梯	341		

第 11 章 游戏中的算法

11.1 洗扑克牌	352
11.1.1 洗扑克牌算法	352
11.1.2 洗扑克牌示例	353
11.2 取火柴游戏	356
11.2.1 取火柴游戏算法	356
11.2.2 取火柴游戏示例	357
11.3 10 点半	358
11.3.1 10 点半算法	358
11.3.2 10 点半游戏示例	363
11.4 生命游戏	368
11.4.1 生命游戏的原理	368
11.4.2 生命游戏的算法	369
11.4.3 生命游戏示例	371
11.5 小结	376

第 3 篇 算法高级应用篇

第 12 章 密码学算法

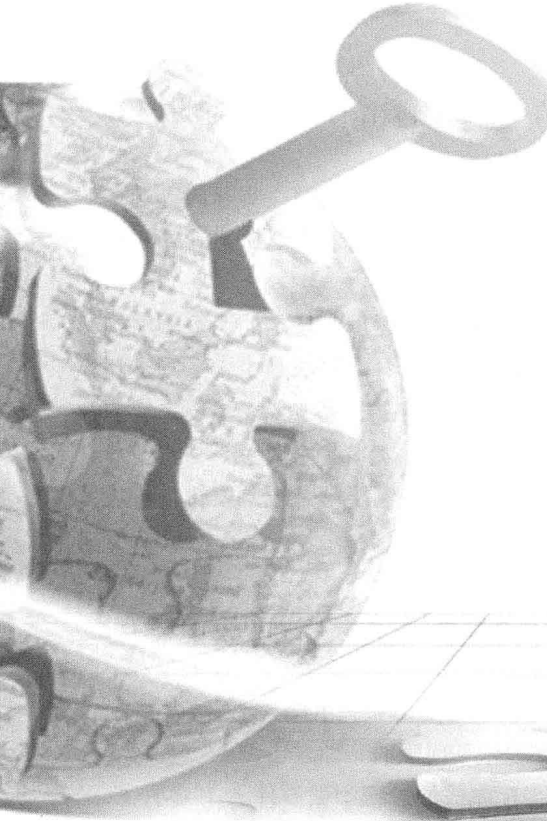
12.1 密码学概述	378	12.1.4 经典密码学算法	380
12.1.1 密码学的发展	378	12.2 换位加密解密	381
12.1.2 密码学的基本概念	379	12.2.1 换位加密解密算法	381
12.1.3 柯克霍夫斯原则	379	12.2.2 换位加密解密算法示例	383
		12.3 替换加密解密	386
		12.3.1 替换加密解密算法	386

- 12.3.2 替换加密解密算法示例 388
- 12.4 位加密解密 389
 - 12.4.1 位加密解密算法 390
 - 12.4.2 位加密解密算法示例 391
- 12.5 一次一密加密解密 392
 - 12.5.1 一次一密加密解密算法 392
 - 12.5.2 一次一密加密解密算法
示例 394
- 12.6 小结 396

第 13 章 压缩与解压缩算法

- 13.1 压缩与解压缩概述 397
 - 13.1.1 压缩与解压缩分类 397
 - 13.1.2 典型的压缩解压缩算法 397
- 13.2 压缩算法 398
- 13.3 解压缩算法 401
- 13.4 压缩/解压缩示例 404
- 13.5 小结 406

第 1 篇 算法基础篇



本篇通过 3 章的篇幅，向读者阐述了算法与数据结构的基本概念和基础思想。前两章着力帮助读者梳理知识思路，搭建概念模型；第 3 章凝聚了作者多年一线开发经验，通过精心归纳和实例演示，深入浅出地阐述了算法的基本思想。

本篇包含：

第 1 章 算法概述

第 2 章 数据结构

第 3 章 基本算法思想

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代码编程交给计算机来完成。在这其中，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

什么是算法（Algorithm）呢？算法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—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，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无穷的规则。

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，算法是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。

可以举一个例子来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3件事呢？一种方法是依次做，如图1-1所示，做完事件A，再做事件B，最后做事件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。

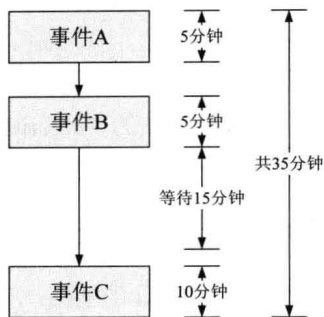


图 1-1 方法一

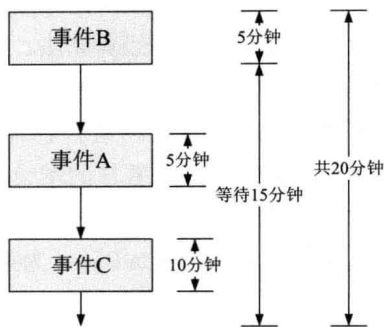


图 1-2 方法二

在上面的例子中，提到了两种方法，可以看做是两种算法。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，算法也是有好坏区分的，好的算法可以提高工作和生活效率。算法的基本任务就是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。

（1）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。

（2）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的任务，每一步做什么事都有明确的规定。

（3）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。

（4）输出

一个算法应该有明确的输出结果。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全部做完了。

（5）可行性

算法的执行步骤必须是可行的，且可以在有限的时间内完成。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，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。

在国外，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”。由于汉字在表述上不太直观，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在算法领域则发展迅速。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，大多数人都认可 Ada Byron 为第一个程序员。他在 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成为可能。算法在计算机程序设计领域再次得到重要的发展。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式才能体现其价值。例如在计算机编程中的算法、数值计算中的算法等。本书重点讲解的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，这里有必要澄清一些基本概念。

1. 算法和公式的关系

前面谈到的算法很容易让我们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，能在有限的时间内完成，并且公式都是完全可以操作并计算的。其实公式确实是提供了一种算法，但算法绝不完全等同于公式。

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。

2. 算法与程序的关系

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，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设计语言编写合适的算法，实现程序的运行。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代码求解问题。

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，他出版了著名《数据结构+算法=程序》一书。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，现在我们对程序、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，则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式。把它们综合起来便构成了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

算法是用来解决实际问题的，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，例如前面的事件 A、B、C 统筹安排的例子。