



妙趣横生 的算法

(C++语言实现)

胡 浩 等编著

6小时教学视频、96个趣味算法题、37个算法面试题，一学就会
帮您开阔眼界，培养编程兴趣，提高编程能力，增强求职竞争力



清华大学出版社

妙趣横生
的算法

(C++语言实现)

胡浩 等编著

清华大学出版社

内 容 简 介

本书以通俗易懂的语言深入浅出地介绍了常用的数据结构和算法知识。在讲解算法理论时，结合实际编程环境，配合精选的例题，利用 C++ 语言将算法理论转换为代码，以加深读者对各种算法的特点和难点的理解，并能更好地掌握算法的应用。另外，作者为书中的重点内容录制了高清教学视频，便于读者高效、直观地学习。

本书共 12 章，分为 4 篇。第 1 篇是基础知识篇，主要介绍常用的数据结构。第 2 篇是基础算法篇，主要介绍算法的一些基础理论和常用的经典基础算法，包含算法的数学基础、排序算法、查找算法等。第 3 篇是高级算法篇，主要介绍难度较高的一些算法，包括高级图算法、动态规划、贪心算法等。其中，高级图算法是重点，用途很广，是在第 3 章图结构的基础上深入分析图结构在工程应用中的作用和常用算法思想，包括拓扑排序和最小生成树等。第 4 篇是算法实战篇，以实例分析为主，包含大量的数学应用方面的算法题和面试中常见的算法题。本篇是对本书算法理论知识的总结和实践，有较高的实用性。

本书特别适合算法入门人员和爱好者阅读，也适合有一定 C++ 编程基础的人员作为进阶读物。另外，本书还适合相关院校作为教材使用。对于参加 IT 企业面试的程序员和各种程序设计选拔赛及信息学竞赛的参赛者，本书也是一本很好的参考读物。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

妙趣横生的算法 (C++ 语言实现) / 胡浩等编著. — 北京：清华大学出版社，2014
ISBN 978-7-302-36760-4

I. ①妙… II. ①胡… III. ①计算机算法 IV. ①TP301.6

中国版本图书馆 CIP 数据核字 (2014) 第 124178 号



责任编辑：夏兆彦

封面设计：欧振旭

责任校对：徐俊伟

责任印制：杨 艳

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：185mm×260mm

印 张：26

字 数：649 千字

版 次：2014 年 10 月第 1 版

印 次：2014 年 10 月第 1 次印刷

印 数：1~3500

定 价：59.80 元

产品编号：056254-01

前言

著名的计算机科学家沃斯(Nikiklaus Wirth)将程序设计形象地通过一个公式总结为“程序=数据结构+算法”。这一公式揭示了程序设计的本质，成为了人们进行程序设计的“黄金定律”。

要想成为一名真正专业的程序员，基本的数据结构基础和常用的算法知识是必须要掌握的。脱离了这两点，其编写出来的程序一定不是健壮的好程序。然而仅仅单纯地掌握一些数据结构基础和常用算法知识也是远远不够的。空洞地掌握所谓的数据结构和算法等理论知识只是纸上谈兵，这些知识必须要依托于程序设计语言才能具有真正的生命力，才能够转化为真实的程序代码，才能够真正地解决实际问题。

本书便是结合 C++ 语言介绍了日常编程实践中常用的数据结构和算法知识，并结合实际给出了大量的编程实例，做到了理论结合实践，让读者在学习算法理论的同时，可以得到 C++ 编程方面的锻炼，从而提高自己的编程实践能力。

本书对每种数据结构和每种算法都秉承先进行概念介绍然后介绍理论，最后进行实例分析的步骤进行。这种循序渐进的方式更能使读者融入算法的世界中，更容易理解书中介绍的算法理论。书中每个实例都经过了实际的编译运行，保证了代码的质量。在介绍每个独立的趣味实例时，将涉及的 C++ 语言中的知识点也都详尽地进行了解说，使读者在解读程序的同时能够对 C++ 语言的常用语法做到融会贯通，牢固掌握。

本书特别适合有一定 C++ 语言基础的读者学习和掌握算法时阅读。为了帮助读者更好、更快、更直观地学习，笔者专门为书中的重点内容录制了高清配套教学视频。这些视频和本书涉及的源程序及其他学习资料一起收录于配书光盘中。

本书特色

1. 注重基础，循序渐进

本书在介绍算法知识时，注重由浅入深的渐进学习方式，注重对基础算法和数据结构的阐述，以夯实基础。在讲解算法实例时，对一些重要的 C++ 语言基础知识点也进行了讲解，这样让读者的算法设计能力和实际编程能力可以同步提高。

2. 实例丰富，生动有趣

本书以通俗易懂的语言，深入细致地介绍了使用 C++ 语言编写的多个算法实例，基本做到了每节都有实例，每种算法都有实例。在介绍实例的同时，将程序开发的基本原理、基本方法和基本技术融入其中。另外，本书选取的算法例题大多都有较强的趣味性，可以

极大地提高读者的学习兴趣，让读者体会到算法学习和程序设计的乐趣。

3. 讲解透彻，注释详尽

本书按照不同类型的趣味问题进行了分类，力求将每一类问题都讲解透彻并总结出解决该类问题的通用和一般的规律。书中的算法实例代码注释详尽，流程图规范，所有的示例均通过测试可以运行，对读者有很好的参考价值。

4. 视频教学，高效直观

本书中的重点内容和实例提供了配套教学视频辅助讲解。读者可以先阅读书中的内容讲解，然后再结合教学视频进行学习，可以获得更加高效而直观的学习效果。

本书主要内容

本书共 12 章，分为 4 篇，主要内容介绍如下。

第1篇 基础知识篇（第1~3章）

第 1 章 基础数据结构，介绍了常用数据结构的理论和应用，为算法学习打下基础。

第 2 章 树结构，介绍了一种非线性数据结构，并用 C++ 将其模型表达出来。

第 3 章 图结构，介绍了图结构的基础知识。

第2篇 基础算法篇（第4~6章）

第 4 章 算法入门，介绍了算法的各种概念和基本思想，为进一步学习打下基础。

第 5 章 排序算法，介绍了编程常用的几种排序算法，并引用了有趣的实例加深读者对排序算法的理解。

第 6 章 查找算法，介绍了常用查找算法的理论和应用，包括折半查找和 B 树查找等。

第3篇 高级算法篇（第7~10章）

第 7 章 高级图算法是对第 3 章图结构的一个升华。本章在图结构的基础上详细探讨了包括拓扑排序、最短路径在内等较难懂却又很实用的图算法。

第 8 章 动态规划，介绍了动态规划算法思想的原理，并引用了大量有趣的实例加以练习，让读者在实践中体会动态规划算法的设计和实现方法。

第 9 章 贪心算法，深入地介绍了贪心算法的理论、特点和具体应用。

第 10 章 综合趣味算法，通过深入浅出的语言，详细介绍了与数学应用有关的几种算法，让读者体会到用算法思想看待数学问题的乐趣。

第4篇 算法实战篇（第11章和第12章）

第 11 章 数学趣题解决算法，通过大量趣味横生的问题和实例，引导读者学会用算法思想看待这些生活难题，并设计出合适的算法加以解决。

第 12 章 面试算法题精粹，通过对精选的数十道名企面试题的详细分析，教会读者

如何从算法的视角去看待和解决这些问题，并设计出可行的解决方案。

本书配套资源获取方式

本书涉及的源程序及教学视频等配套资源需要读者自行下载。请登录清华大学出版社的网站（www.tup.com.cn），搜索到本书页面，按提示下载即可，也可在本书技术论坛上（www.wanjuanchina.net）的相关版式下载。

本书读者对象

- ☐ 算法入门人员；
- ☐ 算法爱好者；
- ☐ 需要提高 C++ 编程水平的人员；
- ☐ IT 企业面试人员；
- ☐ 信息学竞赛的参赛人员；
- ☐ 各种程序设计选拔赛参赛人员；
- ☐ 大中专院校的学生。

本书作者

本书由胡浩主笔编写。其他参与编写的人员有韩先锋、何艳芬、李荣亮、刘德环、孙姗姗、王晓燕、杨平、杨艳艳、袁玉健、张锐、张翔、陈明、邓睿、巩民顺、吉燕、水淼、宗志勇、安静、曹方、曾苗苗、陈超。

如果您在阅读本书的过程中有任何疑问，请发 E-mail 到 bookservice2008@163.com 以获得帮助。

编者

目 录

第 1 篇 基础知识篇

第 1 章 基础数据结构 (📺 教学视频: 26 分钟)	2
1.1 顺序表	2
1.1.1 顺序表的定义	2
1.1.2 顺序表的基本操作	3
1.1.3 实例应用与分析	6
1.2 链表	8
1.2.1 链表的定义	8
1.2.2 链表基本操作	10
1.2.3 双向链表	13
1.2.4 循环链表	17
1.2.5 实例应用与分析	20
1.3 栈	25
1.3.1 栈的定义	25
1.3.2 栈的基本操作	26
1.3.3 实例应用与分析	26
1.4 队列	30
1.4.1 队列的定义	30
1.4.2 队列的基本操作	31
1.4.3 实例应用与分析	31
1.5 散列表	36
1.5.1 散列表的定义	36
1.5.2 散列函数	36
1.5.3 基于散列的查找	37
1.5.4 实例应用与分析	39

第 2 章 树结构 (📺 教学视频: 43 分钟)	43
2.1 树结构简介	43
2.1.1 树的定义	43
2.1.2 树的逻辑表示	44
2.1.3 基本术语	44

2.1.4	树的性质	45
2.1.5	树的基本操作	45
2.1.6	树与森林	46
2.1.7	实例应用与分析	47
2.2	二叉树	49
2.2.1	二叉树的定义	50
2.2.2	二叉树的性质	51
2.2.3	二叉树的存储结构	51
2.2.4	二叉树的基本操作	54
2.2.5	实例应用与分析	64
2.3	线索二叉树	66
2.3.1	线索二叉树的定义	66
2.3.2	线索二叉树的基本操作	67
2.4	二叉查找树	70
2.4.1	二叉查找树的定义	70
2.4.2	二叉查找树的基本操作	71
2.4.3	实例应用与分析	74
2.5	红黑树	78
2.5.1	红黑树的定义	78
2.5.2	红黑树的基本操作	79
2.6	B 树	89
2.6.1	B 树的定义	89
2.6.2	B 树的基本操作	91
第 3 章	图结构 (🔗 教学视频: 34 分钟)	93
3.1	图结构简介	93
3.2	图的存储结构	94
3.2.1	邻接矩阵存储	94
3.2.2	邻接表存储	96
3.3	图的基本操作和图的遍历	97
3.3.1	图的基本操作	97
3.3.2	深度优先遍历	100
3.3.3	广度优先遍历	101
3.3.4	实例应用与分析	102

第 2 篇 基础算法篇

第 4 章	算法入门 (🔗 教学视频: 30 分钟)	106
4.1	什么是算法	106
4.2	算法的数学基础	106
4.2.1	算法分析	106
4.2.2	函数增长率	108

4.3 基本算法思想	108
4.3.1 分治法	108
4.3.2 递归式	109
4.3.3 回溯法	110
4.3.4 穷举法	110
4.3.5 随机算法	111
4.4 基础算法应用实例	111
第 5 章 排序算法 ( 教学视频: 49 分钟)	128
5.1 插入排序	128
5.1.1 算法描述	128
5.1.2 算法实现	129
5.1.3 应用实例与分析	131
5.2 冒泡排序	132
5.2.1 算法描述	132
5.2.2 算法实现	133
5.2.3 应用实例与分析	134
5.3 快速排序	136
5.3.1 算法描述	136
5.3.2 算法实现	137
5.3.3 应用实例与分析	139
5.4 归并排序	141
5.4.1 算法描述	141
5.4.2 算法实现	143
5.4.3 应用实例与分析	145
5.5 希尔排序	146
5.5.1 算法描述	146
5.5.2 算法实现	147
5.5.3 应用实例与分析	148
5.6 选择排序	149
5.6.1 算法描述	149
5.6.2 算法实现	150
5.6.3 应用实例与分析	151
5.7 堆排序	152
5.7.1 堆	152
5.7.2 堆排序算法描述	153
5.7.3 算法实现	154
5.7.4 应用实例与分析	155
5.8 计数排序	158
5.8.1 算法描述	158
5.8.2 算法实现	159
5.8.3 应用实例与分析	160

5.9	基数排序	162
5.9.1	算法描述	162
5.9.2	算法实现	163
5.9.3	应用实例与分析	165
5.10	桶排序	166
5.10.1	算法描述	166
5.10.2	算法实现	168
5.10.3	应用实例与分析	169
5.11	排序算法的比较和选择	172
5.11.1	排序算法的性能比较	172
5.11.2	排序算法的选择标准	172

第 6 章 查找算法 (教学视频：22 分钟)

6.1	顺序查找	174
6.1.1	算法描述	174
6.1.2	算法实现	174
6.1.3	应用实例与分析	175
6.2	二分查找	176
6.2.1	算法描述	176
6.2.2	算法实现	176
6.2.3	实例应用与分析	177
6.3	二叉查找树	178
6.3.1	二叉查找树的定义	178
6.3.2	二叉查找树的基本操作回顾	180
6.3.3	实例应用与分析	183
6.4	分块查找	185
6.4.1	算法描述	185
6.4.2	算法实现	186
6.5	B 树查找算法	187
6.5.1	算法描述	187
6.5.2	算法实现	188
6.5.3	扩展——B+树简介	188
6.6	哈希查找	189
6.6.1	算法描述	189
6.6.2	算法实现	190
6.6.3	实例应用及分析	192

第 3 篇 高级算法篇

第 7 章 高级图算法 (教学视频：30 分钟)

7.1	拓扑排序	196
-----	------	-----

7.1.1	算法描述	196
7.1.2	算法实现	197
7.1.3	实例分析与应用	201
7.2	最小生成树	205
7.2.1	算法描述	206
7.2.2	算法实现	206
7.3	单源最短路径	215
7.3.1	算法描述	215
7.3.2	算法实现	217
7.3.3	应用实例与分析	220
7.4	每对顶点间的最短路径	222
7.4.1	算法描述	222
7.4.2	算法实现	223
7.4.3	应用实例与分析	225
7.5	网络流	225
7.5.1	流网络	225
7.5.2	Ford-Fulkerson 算法描述	227
7.5.3	Ford-Fulkerson 算法的实现	230

第 8 章 动态规划 (📺 教学视频: 17 分钟) 232

8.1	动态规划基础	232
8.1.1	动态规划思想引例	233
8.1.2	动态规划理论基础	236
8.2	动态规划应用实例分析	237
8.2.1	矩阵链乘法问题	237
8.2.2	最长公共子序列问题	240
8.2.3	背包问题	242
8.2.4	求三角形最佳路径问题	246
8.2.5	计算划分方案种类数问题	247
8.2.6	设计竞赛题目问题	248
8.2.7	双人游戏策略问题	249
8.2.8	音乐 CD 最大曲目数问题	250
8.2.9	求 Jerry 最快到达的时间问题	251
8.2.10	陪审团人选问题	254

第 9 章 贪心算法 (📺 教学视频: 21 分钟) 257

9.1	贪心算法概述	257
9.1.1	算法描述	257
9.1.2	算法原理	263
9.2	应用实例与分析	264
9.2.1	活动选择问题	264
9.2.2	任务调度问题	268
9.2.3	闭区间不相交	269

9.2.4 会场安排问题	270
9.2.5 贪心算法在求最小生成树的应用	272

第 10 章 综合趣味算法 ( 教学视频: 34 分钟)	281
10.1 矩阵运算	281
10.1.1 矩阵的性质	281
10.1.2 Strassen 算法求矩阵乘积	285
10.2 字符串算法	289
10.2.1 朴素字符串匹配算法	289
10.2.2 Rabin-Karp 算法	290
10.3 数论有关的基本算法	292
10.3.1 基本数论概念	292
10.3.2 求解最大公约数	293
10.3.3 寻找素数	294
10.3.4 整数的因子分解	297

第 4 篇 算法实战篇

第 11 章 数学趣题解决方法 ( 教学视频: 23 分钟)	304
11.1 求两个数的最大公约数和最小公倍数	304
11.2 魔幻方阵的解法	305
11.3 广场该建多大	309
11.4 水仙花数	312
11.5 通向女友之路	313
11.6 八皇后问题	316
11.7 洗扑克问题	320
11.8 约瑟夫问题	321
11.9 格雷码	325
11.10 老鼠走迷宫	326
11.11 骑士走棋盘	329
11.12 三色球	331
11.13 汉诺塔问题	333
11.14 细胞的游戏	335
11.15 超长整数加法	339
11.16 阿姆斯壮数	342
11.17 筛选求素数	343
11.18 验证角谷猜想	344
11.19 找出说谎人	345
11.20 超级病毒	347
11.21 数字的分解	349

11.22	寻找完美数	350
11.23	二维矩阵转一维矩阵	353
11.24	求解低阶定积分	354
11.25	迭代法开平方运算	357
11.26	牛顿法解方程	358
11.27	求解微分方程	360
11.28	求解线性方程组	361
11.29	梯形法求解定积分	363
第 12 章	面试算法题精粹 ( 教学视频: 24 分钟)	365
12.1	常见基础题精粹	365
12.2	常见算法编程题精粹	372
12.2.1	遍历一次求取单链表的中间点	372
12.2.2	实现单链表的排序	376
12.2.3	判断两棵树是否相等	378
12.2.4	寻找重复率最高的短信	378
12.2.5	实现 strcpy() 函数	382
12.2.6	字符串常见试题	383
12.2.7	实现循环移动	386
12.3	面试进阶问题	389
12.3.1	寻找变位词集合	389
12.3.2	咖啡罐问题	392
12.3.3	判断两个链表是否相交	395
12.3.4	寻找最大的 “N” 个数	396
12.3.5	蜗牛爬杆问题	398
12.3.6	判断单链表里是否有环	399
12.3.7	代码括弧的校验	399
12.3.8	查找数组中的最大值和最小值	401

第 1 篇 基础知识篇

▶▶ 第 1 章 基础数据结构

▶▶ 第 2 章 树结构

▶▶ 第 3 章 图结构

第 1 章 基础数据结构

一直以来，我们谈到算法，必然会同时联想得到数据结构，因为特定的算法需要在相应的数据结构基础上运行才能发挥良好的性能。数据结构是数据的组织形式和载体，而算法是良好的操作步骤，所以，学习数据结构是学好算法的前提。接下来的几章我们会陆续介绍常用的一些数据结构方面的知识。本章先来由浅入深地学习一些基本的数据结构。

1.1 顺 序 表

最基本的数据结构是线性表，线性表被定义为： n 个数据元素的有限序列，其中 n ($n \geq 0$) 表示线性表的长度。顺序表是线性表的一种存储表现形式。下面我们就来详细探讨顺序表。

1.1.1 顺序表的定义

顺序表是最基本的数据结构之一，它属于线性表的一种存储表现形式，在计算机内部表示为一块连续的内存空间，如图 1-1 所示。

根据图 1-1，我们很自然地想到数组结构的示意图与该图非常类似。结合数组的性质可以知道，数组就是一张顺序表，由此可以得出顺序表的一般特性如下：

- ❑ 顺序表的内存空间是一连串连续的地址空间。
- ❑ 与数组需要一个唯一的数组名一样，顺序表也需要有一个唯一的表名来表示，如图 1-1 中的表名 A。
- ❑ 数据在顺序表中按先后顺序排列，存取时可根据相对位置进行随机存取。

顺序表数据结构的 C++定义如下：

```
/******线性表数据结构******/
const int defaultSize = 10;                                     //设置默认顺序表大小
template <typename DataType> class SeqList
{
public:
```

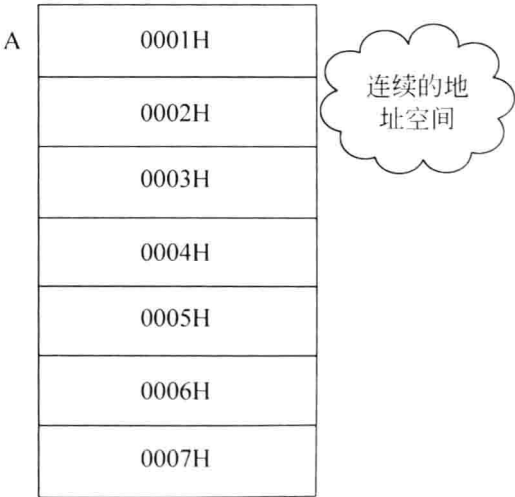


图 1-1 顺序表内存示意图

```

//构造函数
SeqList(int size=defaultSize){
    if(size >0)           //检查赋予的顺序表大小, 如果合法则分配相应大小的内存
    {
        maxSize = size;
        elements = new DataType[maxSize]; //分配内存大小
    }
}
//析构函数
~SeqList()
{
    delete[] elements;           //回收内存空间
}
private:
    Datatype *elements;
    int maxSize;                 //顺序表最大大小
};

```

1.1.2 顺序表的基本操作

顺序表主要有插入元素、删除元素、查找元素、获取元素和修改元素 5 种基本操作。为了方便地实现这 5 种操作, 下面对顺序表的定义做一些修改, 让它能更好地支持这几种操作。

```

/*****线性表数据结构*****/
const int defaultSize = 10;           //设置默认顺序表大小
template <typename DataType> class SeqList
{
public:
    //构造函数
    SeqList(int size=defaultSize){
        if(size >0)           //检查赋予的顺序表大小, 如果合法则分配相应大小的内存
        {
            maxSize = size;
            elements = new DataType[maxSize]; //分配内存大小
        }
    }
    //析构函数
    ~SeqList()
    {
        delete[] elements;           //回收内存空间
    }
    bool insertElement(DataType data);           //向表尾插入新元素
    bool deletElement(int location);             //删除指定位置的元素
    DataType getElement(int location);           //返回指定位置的元素
    bool changeElement(int location,DataType newData); //修改指定位置的元素值
private:
    Datatype *elements;
    int maxSize;                 //顺序表最大大小
};

```

```
int length; //顺序表的有效长度
};
```

1. 顺序表的插入操作

顺序表的插入操作就是向顺序表的表尾插入新的元素，可以分为两个步骤来进行。

- (1) 检查顺序表是否已满，如果已满则进行提示并拒绝执行插入操作。
- (2) 如果顺序表未满，则将新元素插入表尾空间中，并将顺序表的有效大小加1。

顺序表的插入操作代码如下：

```
/******顺序表插入操作*****/
template <typename DataType> bool SeqList<DataType>::insertElement
(DataType data)
{
    int currentIndex = length; //记录新元素的插入位置
    if(length >= maxSize) //判断顺序表是否已满
    {
        return false; //顺序表已满，返回 false，插入不成功
    }
    else
    {
        elements[currentIndex] = data; //将新元素插入顺序表表尾
        length++; //顺序表有效长度加1
        return true;
    }
}
```

在上面的程序中，让插入函数返回一个 bool 型变量，以反馈插入操作是否成功。当然，很多书籍中也将这类函数设置为无返回函数，即返回 void，但是笔者认为返回 boolean 型数据更有利于程序运行的错误检查，所以在后面的程序中，我们会尽量使这类操作有 bool 型返回值，用以大型程序运行结果的检查。

2. 顺序表的删除元素操作

顺序表的删除元素操作主要完成删除顺序表中指定元素的操作，所以需要为操作函数提供一个元素所在位置的索引号作为参数。例如，要删除顺序表 A 的位置 i 上的元素，即

$A(a_1, a_2, a_3, a_4 \dots a_{i-1}, a_i, a_{i+1} \dots a_n)$

我们要删除第 i 个位置上的元素 a_i ，删除后顺序表应变为：

$A(a_1, a_2, a_3, a_4 \dots a_{i-1}, a_{i+1} \dots a_n)$

此时，顺序表第 i 个位置后面的元素一次前移补充空出的位置，同时顺序表的长度减1。

```
/******删除指定位置的元素*****/
template <typename DataType> bool SeqList<DataType>::deleteElement(int
```