

ART

OF

ALGORITHM

DESIGN

算法设计艺术

王秋芬 刘平 杜鹃 著

ART

OF

ALGORITHM

DESIGN

ART

OF

ALGORITHM

DESIGN

清华大学出版社

算法设计艺术

王秋芬 刘平 杜鹃 著

藏书

清华大学出版社

北京

内 容 简 介

本书采用图文并茂的方式将抽象的知识点具体化；引用生活中各种各样的趣味故事与趣味题目引导、类比数据结构和算法中的知识点，增强读者学习的趣味性；对大多数算法用具体案例逐步剖析，对算法描述的关键语句进行详细注解，增强算法的可理解性。本书将数据结构知识和算法知识进行有机融合，本书内容新颖，可读性强，易于理解。

本书主要介绍算法的渊源、算法的作用、基本算法知识和数据结构知识、多种算法设计思想在排序、选择问题及图算法中的应用，经典算法与智能计算方法的融合改进算法可使读者在学习数据结构和算法的同时，从不同角度认识算法与理解数据结构，学会用发散思维方式思考问题。

本书适合学过一门计算机高级程序设计语言的各类读者，包括在校计算机专业的大学生、业余编程爱好者、程序员等。

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

算法设计艺术/王秋芬,刘平,杜娟著.--北京:清华大学出版社,2014

ISBN 978-7-302-36745-1

I. ①算… II. ①王… ②刘… ③杜… III. ①电子计算机—算法设计 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2014)第 124174 号

责任编辑：梁 颖

封面设计：常雪影

责任校对：焦丽丽

责任印制：宋 林

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

课 件 下 载：<http://www.tup.com.cn>, 010-62795954

印 刷 者：北京富博印刷有限公司

装 订 者：北京市密云县京文制本装订厂

经 销：全国新华书店

开 本：185mm×230mm 印 张：13.75 字 数：302 千字

版 次：2014 年 6 月第 1 版 印 次：2014 年 6 月第 1 次印刷

印 数：1~1000

定 价：39.00 元

前 言

一、本书起因

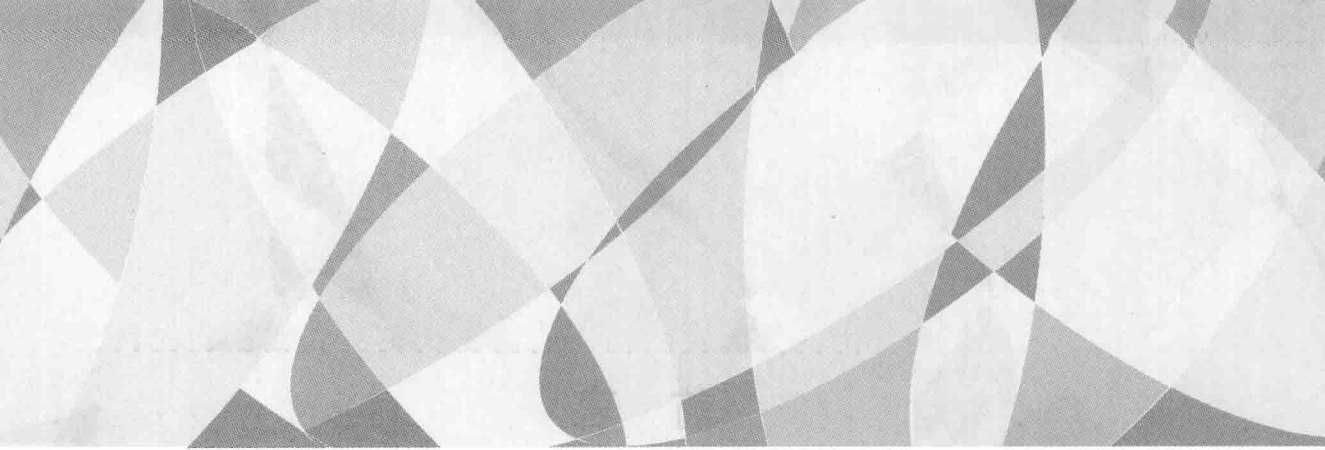
喜欢编程的读者都知道,程序设计不仅需要娴熟地掌握一门程序设计语言,更重要的是数据结构的选择和算法的设计。因此,数据结构和算法设计这两部分知识对程序设计者都非常重要。所有的学习者都认识到这一点,但也有共同的感慨:数据结构抽象,算法理解困难。究其原因,笔者认为这两门课程有着千丝万缕的联系,而传授这两部分知识时却相互分离,缺少穿插。算法设计前必须选择合适的数据结构,数据结构选择时也要适当考虑算法设计的方便性和有效性,这两部分知识相辅相成。因此,笔者将这两部分知识进行融合,撰写了《算法设计艺术》一书,从算法策略的角度来理解数据结构中的算法,从算法思想上理解数据结构的选择,不同的数据结构又直接导致算法设计的差异,同时本书还重点考虑了知识的趣味性和前沿性。

二、本书结构

本书内容共分为6章,其中第1章介绍算法基础知识;第2章介绍常用数据结构知识;第3章从算法设计策略角度讲解多种排序算法;第4章介绍选择算法;第5章介绍图算法;第6章介绍遗传算法及其扩展。具体结构安排如下:

第1章基本算法知识,主要介绍算法的由来、算法届大师、算法在计算中的作用、算法渐进性分析、算法的重要性、基本算法设计策略等。

第2章基本数据结构,主要介绍了线性表、树、图、集合4种类型



的数据结构。

第3章排序算法,由十二生肖排序的故事引出排序的基本概念,从算法策略的角度讲解多种排序算法:贪心排序算法、分治排序算法、搜索排序算法、随机排序算法、基于模运算的排序算法、分组排序算法、位排序算法等。

第4章选择算法,主要介绍了多种找最小值最大值的算法、找中位数的随机算法和线性时间选择算法。

第5章图算法,主要介绍图遍历算法、最短路径算法、最小生成树算法和二分图算法等。

第6章遗传算法及其拓展,主要介绍了遗传算法的相关理论,结合0-1背包问题、旅行商问题讲解遗传算法与经典算法思想的融合拓展设计的新算法,以引发读者进一步学习和研究的兴趣。

南阳理工学院王秋芬撰写第1章和第3章,同时负责全书的统稿与审校工作,南阳理工学院刘平撰写第2章和第5章,南阳理工学院杜娟撰写第4章、第6章和参考文献,并参与书稿的审校工作。

三、本书特色

1. 图文并茂

俗话说“一图值千金”,一篇文章如果加上了插图,它的效果是可想而知的。尤其是科技类文章,插图有助于简明扼要地向读者传递信息,起到文字不能替代的作用。本书针对学习者共同的困惑——数据结构抽象,算法理解困难,在撰写抽象数据结构知识时插入了丰富的精美图片,用一种相对具体的方式来诠释这些抽象的知识,增强它们的可理解性,易于掌握;在算法描述中,采用了大量形象直观的



图片解说算法执行的过程,逐步分解剖析,降低算法理解的难度。这些精美图片与知识点文字相互陪衬,相得益彰,使读者一目了然,使读者在学习数据结构和算法的过程中事半功倍。

2. 趣味引领

无论是数据结构知识还是算法知识,大部分图书在内容上都是直奔主题,学习者学起来觉得抽象、难懂。但是如果让学习者通过已经掌握的知识或司空见惯的生活实践来理解刚刚接触的新知识,效果会很好。例如,生活中有很多排序的情境,教师喜欢按照学生成绩升序排序(或按学号排序),销售部门经常按照员工业绩排序,超市按期统计各种商品的销量并按照销量排序等,数据结构中的排序算法也正是生活中排序方法的归纳和总结,以此引导学生思考,激发学习积极性和主动性;再如,笔者用幼儿园中小朋友的活动场景来解说线性表的概念,用银行排队、旅游排队、学生食堂排队等现象来阐述队列的由来和形式。因此,本书引用了很多生活中大家熟知的内容,引导学生学习新知识,使学习者在学习的过程中充满趣味、享受快乐。

3. 案例详解

为了增强本书的可读性和易理解性,本书针对书中的绝大部分算法,都给出了具体案例,演示算法的执行过程。同时配备了与算法各个执行步骤相一致的文字和图片,让算法抽象的执行过程具体化,便于学习者接受、理解。在用程序设计语言描述算法时,算法的大部分语句都有详细的注释,帮助学习者理解算法语句的含义。本书这一特色将对读者学习数据结构和算法有很大帮助。



四、本书读者

数据结构和算法是计算机软件专业的基础必修课,可以说,要想从事程序设计工作,必须熟练掌握这些知识。因此,本书的读者非常广泛,包括在校本专科计算机专业及其相关专业的学生、编程爱好者、各类想从事计算机编程工作的专业或非专业人士。

在此,谨向清华大学出版社负责本书编辑出版工作的全体同仁和每一位关心和支持本书撰写工作的各方面专家表示衷心感谢。

由于笔者水平有限,书稿虽几经修改,仍难免有疏漏或不妥之处,欢迎广大读者和专家批评指正。联系方法:电子邮件 w_qiufen@163.com。

作者

2014年1月

目 录

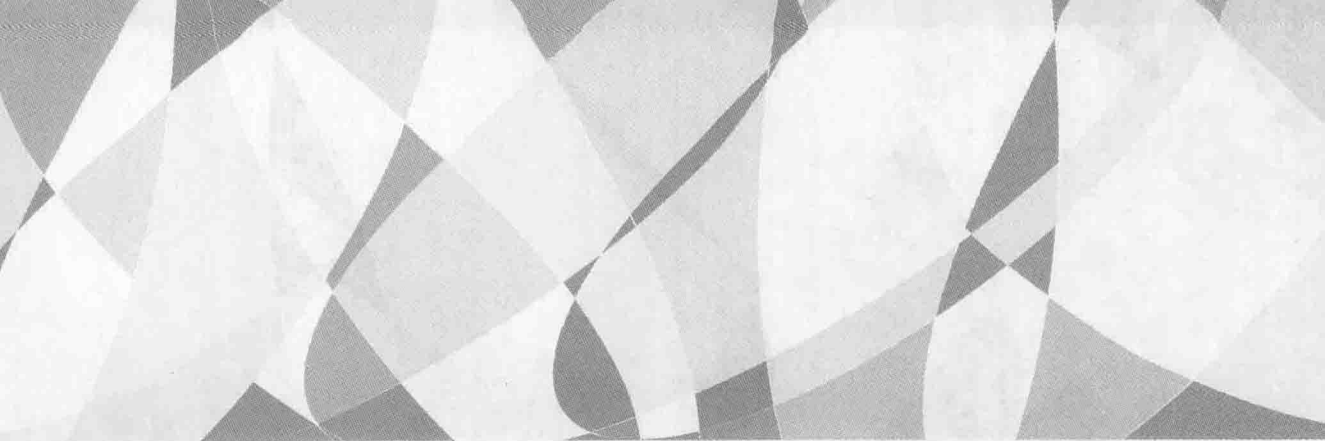
第 1 章 基本算法知识	1
1.1 开场白	1
1.2 “算法”的由来	2
1.3 算法十大名师	3
1.4 算法在计算中的作用	7
1.4.1 算法及其特性	7
1.4.2 算法的应用	9
1.5 算法渐近性分析	11
1.6 学习算法的重要性	13
1.7 基本算法设计策略	14
1.7.1 贪心法	14
1.7.2 分治法	17
1.7.3 回溯法	19
1.7.4 分支限界法	23
1.7.5 随机化算法	25
1.7.6 动态规划	29
习题	31
第 2 章 基本数据结构	33
2.1 开场白	33
2.2 线性表	33
2.2.1 顺序表和链表	35
2.2.2 栈与队列	41
2.3 树	52

2.3.1	树的定义	52
2.3.2	树结构中的重要术语	53
2.3.3	树的存储结构	55
2.3.4	最优二叉树(哈夫曼树)	57
2.3.5	最优二叉搜索树	62
2.4	图	69
2.5	集合	74
	习题	75

第3章 排序算法 77

3.1	十二生肖排序的故事	77
3.2	排序的基本概念	79
3.3	贪心排序	81
3.4	分治排序	95
3.4.1	递归算法	95
3.4.2	分治排序算法	98
3.5	搜索排序	104
3.5.1	二叉树的定义及遍历	104
3.5.2	二叉搜索树	111
3.5.3	二叉搜索树排序	117
3.6	随机排序	118
3.7	基于模运算的排序	120
3.8	分组排序	126
3.9	位排序	130
	习题	134

第 4 章 选择算法	136
4.1 最小值与最大值	136
4.2 中位数选择	140
4.3 线性时间选择	142
4.3.1 随机线性时间选择	142
4.3.2 分组线性时间选择	145
习题	148
第 5 章 图算法	150
5.1 图的遍历算法	150
5.1.1 深度优先遍历	150
5.1.2 广度优先遍历	155
5.2 单源最短路径算法	160
5.3 最小生成树算法	168
5.4 二分图算法	174
5.4.1 二分图概念	174
5.4.2 最大流算法	175
5.4.3 匈牙利算法	184
习题	189
第 6 章 算法拓展	192
6.1 遗传算法	192
6.2 贪心遗传算法	196



6.3 启发式遗传算法 200

习题 207

参考文献 208

第 1 章 基本算法知识

1.1 开场白

编程爱好者都酷爱编程,但编程的实质是什么并不是所有人都清楚。Niklaus Wirth 给出了描述程序设计、数据结构和算法之间关系的著名公式:“算法+数据结构=程序”。数据结构描述了非数值问题的程序设计中数据对象以及它们之间的关系、相关操作,它是程序设计的基础;算法是计算机解决非数值问题的方法步骤的描述,是程序设计的精髓。可见,算法是程序设计的主要组成部分。

作为计算机程序的主要组成部分,算法是强大的,已经影响到我们的生活和工作,例如,上淘宝网购物、逛唯品会已成为很多女士生活中不可缺少的一部分;游戏备受年轻人的青睐,有的还达到痴迷的程度;公安干警通过网络获取破案线索、追捕逃犯;股民通过电脑、网络关注股市行情;购物者可以根据运单号进行快递跟踪;人们可以网上购票,用办公自动化软件进行办公,用各种管理软件进行管理;这些多样的应用背后,算法正改变着我们的生活方式。

算法是隐蔽的,能给人们带来便利。习惯用 Excel 管理数据的人都熟悉这样的操作:需要对数据简单求和时,只需要选中求和的数据,然后点击工具栏上的求和按钮即可;需要筛选、排序数据时,只需要操作相关菜单项即可;需要求算术平均时稍微麻烦一些,插入求平均的函数,然后选择需要求平均的数据就可以了;很多关于数据的操作都非常方便,岂不知这些便利都是算法带来的。再如,登录淘宝网购物时,琳琅满目的商品让人眼花缭乱,但是,我们可以通过搜索、信用排序、折扣排序、销量排序等操作快速将感兴趣的商品找出来,这些操作的背后运行着算法;当浏览某种商品时,网页底部同样给出喜欢(浏览)该商品的人还喜欢(浏览)的商品,它们会影响很多人的决定,给销售商带来很大利润,这同样也是算法带来的。

算法给人们带来便利的同时也会带来危害,如纽约证券经纪交易公司骑士资本的一台电脑上的“流氓程序”自动进入自动交易系统,使得该公司亏损 4 亿 4 千万美金。再如计算

机病毒带来的损失，“爱丽莎”病毒和“探测者”病毒分别造成了 12 亿和 10 亿美元的经济损失；“爱虫”病毒及其以后出现的 50 来个变种病毒迄今为止共感染了 4000 多万台计算机，造成大约 87 亿美元的经济损失；互联网上引起大规模恐慌的“红色代码”病毒所造成的损失大约接近 26 亿美元。病毒有很强的隐蔽性，发现时可能已经造成了较大的损失。

算法不仅在商业方面大显神威，而且能左右人们的思维方式。现在人们购物不仅仅是现金交易，还热衷于刷卡；手机缴费、转账、还信用卡等不再需要亲自去相应的营业厅办理，可以登录网络银行完成；逛商场时看到喜欢的商品，很多年轻人喜欢“扫一扫”获知该商品详情；去图书馆借书，首先通过图书查询系统查找想借的书籍是否存在、在馆，若在馆，则记录陈列位置，然后直奔相应位置将书借出，一改往日盲目性；想了解一个新事物或新名词时，首先“百度一下”或上谷歌搜索等。这些生活中思维方式的改变无不是算法带来的。

搜索是人们常用的操作，在淘宝网搜索关心的商品，在当当、亚马逊搜索想要的图书，在百度、谷歌等搜索关心的话题等。搜索到的相关内容很丰富，但人们习惯性地关注排在前面的内容，而排在后面的几乎无人问津。能否被人们看到、了解直接影响着商家的经济效益。基于这样的事实，所有的商品供应商、所有网站都希望自己能被搜索到且排在首页，这与搜索引擎算法密不可分。搜索引擎的竞争实质上是搜索引擎算法的优劣。网页排名算法是搜索引擎算法中的子算法，在该算法中，一个网站被其他网站链接的质量和数量等，都被认为是网站在搜索结果排名的重要因素，不过各个搜索引擎的网页排名算法对于网站外部链接的质量以及其他因素的权重可能有一定的差异。各大网站为了能被网页排名算法排在首页，纷纷在符合用户友好性及搜索引擎算法的基础上，使用网站内或者网站外的优化手段，使网站在搜索引擎的关键词排名提高，从而获得目标搜索流量，提高网页排名。

网页排名使得谷歌无比强大，但也给各国监管机构带来了严重问题。一方面，该算法是受到严格保护的商业机密，如果不受到保护，则搜索引擎优化会被滥用，人们也不再信任搜索结果；另一方面，谁也不能保证谷歌不会为了自己的利益而干扰搜索结果。

由此，在信息化的今天，算法无时无刻不在人们周围。摆在每个人面前的事实就是：“要么编程，要么被动接受编程”，这足以显示算法的影响和威力。

1.2 “算法”的由来*

“算法”的大陆中文名称出自周髀算经；而英文名称 Algorithm 来自于 9 世纪波斯数学家比阿勒·霍瓦里松(阿尔·花拉子米)的名字 al-Khwarizmi，因为比阿勒·霍瓦里松在数学上提出了算法这个概念。Algorithm(算法)一词本身就十分有趣。初看起来，这个词好像是某人打算要写“Logarithm”(对数)一词但却把头四个字母写的前后颠倒了。这个词一直

* 本节部分内容参考了文献[36]。

到 1957 年之前在 *Webster's New World Dictionary* (《韦氏新世界词典》) 中还未出现, 我们只能找到带有它的古代含义的较老形式的“Algorism”(算术), 指的是用阿拉伯数字进行算术运算的过程。在中世纪时, 珠算家用算盘进行计算, 而算术家用算术进行计算。中世纪之后, 对这个词的起源已经拿不准了。早期的语言学家试图推断它的来历, 认为它是从把 *algiros* (费力的) + *arithmos* (数字) 组合起来派生而成的; 但另一些人则不同意这种说法, 认为这个词是从“喀斯迪尔国王 *Algor*”派生而来的。

最后, 数学史学家发现了 *algorism* (算术) 一词的真实起源: 它来源于著名的 *Persian Textbook* (《波斯教科书》) 的作者的名字 *Abu Ja'far Mohammed ibn Mūsā al-Khowārizm* (约公元前 825 年)——从字面上看, 这个名字的意思是“*Ja'far* 的父亲, *Mohammed* 和 *Mūsā* 的儿子, *Khowārizm* 的本地人”。*Khowārizm* 是前苏联 *ХИВА* (基发) 的小城镇。*Al-Khowārizm* 写了著名的书 *Kitab al jabr w'al-muqabala* (《复原和化简的规则》); 另一个词, “*algebra*”(代数), 是从他的书的标题引出来的, 尽管这本书实际上根本不是讲代数的。

逐渐地, “*algorism*”的形式和意义就变得面目全非了。如牛津英语字典所说的, 这个词是由于同 *arithmetic* (算术) 相混淆而形成的错拼词, 由 *algorism* 又变成 *algorithm*。一本早期的德文数学词典 *Vollständiges Mathematisches Lexicon* (《数学大全辞典》), 给出了 *Algorithmus* (算法) 一词的如下定义: “在这个名称之下, 组合了四种类型的算术计算的概念, 即加法、乘法、减法、除法”。拉丁短语 *algorithmus infinitesimalis* (无限小方法), 在当时就用来表示 *Leibnitz* (莱布尼兹) 所发明的以无限小量进行计算的微积分方法。

1950 年左右, *algorithm* 一词经常同欧几里得算法 (*Euclid's algorithm*) 联系在一起。这个算法就是在欧几里得的《几何原本》(*Euclid's Elements*, 第 VII 卷, 命题 i 和 ii) 中所阐述的求两个数的最大公约数的过程 (即辗转相除法)。欧几里得算法被人们认为是史上第一个算法。第一次编写程序是 *Ada Byron* 于 1842 年为巴贝奇分析机编写求解伯努利方程的程序, 因此 *Ada Byron* 被大多数人认为是世界上第一位程序员。因为查尔斯·巴贝奇 (*Charles Babbage*) 未能完成他的巴贝奇分析机, 这个算法未能在巴贝奇分析机上执行。因为 “*well-defined procedure*” 缺少数学上精确的定义, 19 世纪和 20 世纪早期的数学家、逻辑学家在定义算法上出现了困难。20 世纪的英国数学家图灵提出了著名的图灵论题, 并提出一种假想的计算机的抽象模型, 这个模型被称为图灵机。图灵机的出现解决了算法定义的难题, 图灵的思想对算法的发展起到了重要的作用。

1.3 算法十大名师

1. 伟大的智者——*Donald E. Knuth* (高纳德 E. 克努斯)

Donald E. Knuth 生于 1938 年, 是算法和程序设计技术的先驱者。他被公认是美国最聪明的人之一。当年他上大学的时候, 常编写各种各样的编译器来挣外快。只要是他参加的编程比赛, 第一名非他莫属。他也是世上少有的编程达到 40 年以上的程序员之一。

他除了是技术与科学上的泰斗外，也是一位写作高手，技术文章堪称一绝。他的文章文风细腻，讲解透彻，思路清晰而且没有学究气，其经典著作《计算机程序设计艺术》被誉为算法中的圣经。在该著作中，像 KMP 和 LR(K) 等令人不可思议的算法比比皆是。

Donald E. Knuth 一生中获得的奖项和荣誉不计其数，包括图灵奖、美国国家科学金奖、美国数学学会斯蒂尔奖 (AMS Steel Prize)，及发明先进技术荣获的极受尊重的京都奖 (Kyoto Prize) 等。他写过 19 部书和 160 余篇论文，每一篇都有深远的影响。

Donald E. Knuth 维基百科链接：http://en.wikipedia.org/wiki/Donald_E_Knuth。

2. 首席算法官——Udi Manber(乌迪·曼博)

Udi Manber 是著名的美国计算机科学家。他主要研究 WWW 的应用程序、搜索以及隐藏在这背后的算法设计，并与其他人共同开发了 Agrep、Glimpse 和 Harvest 等 UNIX 上的搜索软件。1998 年，Udi 成为了 Yahoo 的首席科学家。2002 年，Amazon 创造性地给了 Udi“首席算法官”的职位，和 Udi 为 Amazon 的 Search Inside the Book 搜索项目所做的工作相得益彰。其著作 *Introduction to Algorithms—A Creative Approach* 被大家称道。

Udi Manber 提出了一种搜索引擎质量评估标准 UDI test。其主要思想是：评价搜索引擎的质量主要看用户的体验，由普通用户在没有任何暗示的情况下对两个搜索引擎的结果进行等级评分，并且在这种评分中引入用户信心指数，最后通过对大量评分的统计来得到两个搜索引擎的质量评分。用来做 UDI test 的关键词一般需要庞大的数量，这些关键词通常从检索日志中随机提取得到，非常具有代表性。UDI test 的优点在于在评测搜索引擎质量时引入用户体验作为重要的评测参数，使得测试效果更加有效。

Udi Manber 维基百科链接：http://en.wikipedia.org/wiki/Udi_Manber。

3. 谦逊的长者——Edsger Wybe Dijkstra(艾兹格·W. 迪科斯彻)

E. W. Dijkstra(1930—2002) 与 D. E. Knuth 并称为这个时代最伟大的计算机科学家。他在算法、操作系统、分布式处理等许多方面都有极高深的造诣。除了科学研究之外，他最喜欢教学，被称作“一天教学 24 小时”的教授。

他在祖国荷兰获得数据和物理学学士、理论物理博士学位。2000 年退休前，一直是美国 Texas 大学的计算机科学和数学教授。1959 年，他发现了图论中的最短路径算法 (Dijkstra 算法)，并因此而闻名于世；他提出了“goto 有害论”和信号量及 PV 原语，并解决了“哲学家聚餐”问题，同时设计开发了 THE 操作系统。1972 年，他因 ALGOL 第二代编程语言而获得图灵奖。他在获得图灵奖时的演讲中仍然称自己不过是一个谦逊普通的程序员，其经典著作 *Go To Statement Considered Harmful (EWD215)* 至今仍被人们广为传颂。

Edsger Wybe Dijkstra 经典言论：

- 编程的艺术就是处理复杂性的艺术。
- 优秀的程序员很清楚自己的能力是有限的，所以他对待编程任务的态度是完全谦卑的，特别是，他们会像逃避瘟疫那样逃避“聪明的技巧”。

- 计算机科学是应用数学最难的一个分支,所以如果你是一位蹩脚的数学家,最好留在原地,继续当你的数学家。
- 我们所使用的工具深刻地影响我们的思考习惯,从而也影响了我们的思考能力。
- 实际上如果一个程序员先学了 BASIC,那就很难教会他好的编程技术了,因为作为一个可能的程序员,他们的神经已经错乱了,而且无法康复。
- 就语言的使用问题,根本不可能用一把钝斧子削好铅笔,而换成十把钝斧子会使事情变成大灾难。
- 简单是可靠的先决条件。

Edsger Wybe Dijkstra 维基百科链接: http://en.wikipedia.org/wiki/Edsger_Wybe_Dijkstra。

4. 运筹学大师——George Dantzig(乔治·丹茨格)

美国数学家 George Bernard Dantzig 生于 1914 年 11 月 8 日,于 2005 年 5 月 13 日逝世。1947 年,33 岁的 Dantzig (丹茨格)提出了解决一种最优化问题的单纯形法,该方法奠定了线性规划的基础,使得经济学、环境科学、统计学应用等学科获得了迅速发展。Dantzig 也因而被誉为“线性规划之父”。

后来,Dantzig 在运筹学领域中的建树极高,获得了包括“冯·诺伊曼理论奖”在内的诸多奖项。他在 *Linear programming and extensions* 一书中研究了线性编程模型,为计算机语言的发展做出了不可磨灭的贡献。

George Dantzig 维基百科链接: http://en.wikipedia.org/wiki/George_Dantzig。

5. 推动时代前进的人——James Cooley(詹姆斯·库利)

James Cooley 生于 1926 年,美国数学家,哥伦比亚大学的数学博士。他和普林斯顿大学的 John Tukey,AT&T 贝尔实验室共同创造的快速傅立叶变换(FFT)意义极其重大。FFT 的数学意义不光在于使大家明白了傅立叶(Fourier)变换计算起来是多么容易,而且使得数字信号处理技术取得了突破性的进展,为现在的网络通信,图形图像处理等领域的发展与前进奠定了基础。Fourier 变换的意义在于将电能变为了工业的命脉,而 FFT 的意义更是在于它推动了整个社会信息化的进程。

在 1992 年退休之前,James Cooley 一直在 IBM 研究中心从事数字信号处理的研究,同时他还是 IEEE 数字信号处理委员会的成员。1980 年获得 ASSP's Meritorious Service Award,1984 年获得 ASSP Society Award 以及 IEEE Centennial Medal。

James Cooley 维基百科链接: http://en.wikipedia.org/wiki/James_Cooley。

6. FORTRAN 之父——John W. Backus(约翰·巴库斯)

早年,John W. Backus 在宾夕法尼亚州著名的 Hill 中学求学时,讨厌学习,成绩一塌糊涂。18 岁时,他在父亲的要求下到弗吉尼亚大学学习化学,但他每周只上一堂音乐欣赏课,最终被学校开除;随后参过军、在哈弗福德医学院学习,度过了一段混沌岁月。战后 John W. Backus 进入纽约哥伦比亚大学学习数学,在 1950 年毕业后进入 IBM 计算机中心,成为

了一名程序员。

在 IBM, John W. Backus 的才华得到了施展, 发明了人类历史上第一个高级语言——FORTRAN。接着, 又提出了规范描述编程语言——巴克斯-诺尔范式(BNF)。1977 年, 美国计算机协会于授予 John W. Backus 图灵奖, 他在颁奖典礼上发表的演说题目是 Can Programming Be Liberated From the von Neumann Style(程序设计能脱离冯诺依曼风格吗?)。

John W. Backus 维基百科链接: http://en.wikipedia.org/wiki/John_Backus。

7. 实践探索先锋——Jon Bentley(乔恩·本特利)

Jon Bentley 于 1974 年获得了斯坦福大学的学士学位, 1976 年获得北卡罗莱纳大学的硕士和博士学位。毕业后在卡耐基梅隆大学教授了 6 年计算机科学课程, 1982 年进入贝尔实验室, 2001 年退休后加入了现在的 Avaya 实验室。他还曾作为访问学者在西点军校和普林斯顿大学工作。他的研究领域包括编程技术、算法设计、软件工具和界面设计等。

Jon Bentley 写作过 3 本编程书籍, 其中最著名的就是涵盖从算法理论到软件工程各种主题的 *Programming Pearls*(编程珠玑)。在这本书中, Jon Bentley 从工程实现的角度出发, 为那些艰难的问题提供了独特而巧妙的解决方案, 这些解决方案犹如一颗颗闪闪发亮的珍珠。

Jon Bentley 维基百科链接: http://en.wikipedia.org/wiki/Jon_Bentley。

8. Pascal 之父——Niklaus Wirth(尼克劳斯·威茨)

Niklaus Wirth 于 1934 年出生于瑞士, 1963 年在加州大学伯克利分校取得博士学位。其后直接被以高门槛著称的斯坦福大学聘到刚成立的计算机科学系工作。在斯坦福大学成功地开发出 Algol W 以及 PL360 后, 于 1967 年回到祖国瑞士, 第二年在他的母校苏黎世工学院创建与实现了 Pascal 语言。Pascal 语言是当时最流行的语言, 他的学生 Philippe Kahn 和 Anders Hejlsberg 创办的 Borland 公司靠 Turbo Pascal 起家, 并很快将 Borland 发展成为全球最大的开发公司, 这一切都归功于 Pascal 语言的魅力。Pascal 已经影响了整整几代程序员, Niklaus Wirth 的思想还将继续为现在和以后的程序员指引前进的方向。

1971 年, Niklaus Wirth 基于其开发程序设计语言和编程的实践经验, 在 4 月份的 *Communications of ACM* 上发表了论文“通过逐步求精方式开发程序”(Program Development by Stepwise Refinement), 首次提出了“结构化程序设计”(structure programming)的概念。20 世纪 70 年代中期, 为适应并发程序设计的需要, Niklaus Wirth 又成功开发了一个获得广泛应用的语言 Modula。1984 年, Niklaus Wirth 获得了图灵奖, 提出了著名的公式“算法+数据结构=程序”。

Nicklaus Wirth 维基百科链接: http://en.wikipedia.org/wiki/Niklaus_Wirth。

9. 算法的讲解者——Robert Sedgwick(罗伯特·赛奇威克)

Robert Sedgwick 在斯坦福大学获得博士学位, 是普林斯顿大学的计算机科学教授, 还是 Adobe Systems 的一名主管, 也曾作为访问学者在 Xerox PARC、IDA 和 INRIA 工作。