

算法的乐趣 (图灵原创)

作者：王晓华

版权信息

书名：算法的乐趣

作者：王晓华

ISBN：978-7-115-38537-6

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

图灵社区会员 ptpress (libowen@ptpress.com.cn) 专享 尊重版权

[序一](#)

[序二](#)

[序三](#)

[致谢](#)

[前言](#)

[第1章 程序员与算法](#)

[1.1 什么是算法](#)

[1.2 程序员必须要会算法吗](#)

[1.2.1 一个队列引发的惨案](#)

[1.2.2 我的第一个算法](#)

[1.3 算法的乐趣在哪里](#)

[1.4 算法与代码](#)

[1.5 总结](#)

[1.6 参考资料](#)

[第2章 算法设计的基础](#)

[2.1 程序的基本结构](#)

[2.1.1 顺序执行](#)

[2.1.2 循环结构](#)

[2.1.3 分支和跳转结构](#)

[2.2 算法实现与数据结构](#)

[2.2.1 基本数据结构在算法设计中的应用](#)

[2.2.2 复杂数据结构在算法设计中的应用](#)

[2.3 数据结构和数学模型与算法的关系](#)

[2.4 总结](#)

[2.5 参考资料](#)

[第3章 算法设计的常用思想](#)

[3.1 贪婪法](#)

[3.1.1 贪婪法的基本思想](#)

[3.1.2 贪婪法的例子：0-1背包问题](#)

[3.2 分治法](#)

[3.2.1 分治法的基本思想](#)

[3.2.2 递归和分治，一对好朋友](#)

[3.2.3 分治法的例子：大整数Karatsuba乘法算法](#)

[3.3 动态规划](#)

[3.3.1 动态规划的基本思想](#)

[3.3.2 动态规划法的例子：字符串的编辑距离](#)

[3.4 解空间的穷举搜索](#)

[3.4.1 解空间的定义](#)

[3.4.2 穷举解空间的策略](#)

[3.4.3 穷举搜索的例子：Google方程式](#)

[3.5 总结](#)

[3.6 参考资料](#)

[第4章 阿拉伯数字与中文数字](#)

[4.1 中文数字的特点](#)

[4.1.1 中文数字的权位和小节](#)

[4.1.2 中文数字的零](#)

[4.2 阿拉伯数字转中文数字](#)

[4.2.1 一个转换示例](#)

[4.2.2 转换算法设计](#)

[4.2.3 算法实现](#)

[4.2.4 中文大写数字](#)

[4.3 中文数字转阿拉伯数字](#)

[4.3.1 转换的基本方法](#)

[4.3.2 算法实现](#)

[4.4 数字转换的测试用例](#)

[4.5 总结](#)

[4.6 参考资料](#)

[第5章 三个水桶等分8升水的问题](#)

[5.1 问题与求解思路](#)

[5.2 建立数学模型](#)

[5.2.1 状态的数学模型与状态树](#)

[5.2.2 倒水动作的数学模型](#)

[5.3 搜索算法](#)

[5.3.1 状态树的遍历](#)

[5.3.2 剪枝和重复状态判断](#)

[5.4 算法实现](#)

[5.5 总结](#)

[5.6 参考资料](#)

[第6章 妖怪与和尚过河问题](#)

[6.1 问题与求解思路](#)

[6.2 建立数学模型](#)

[6.2.1 状态的数学模型与状态树](#)

[6.2.2 过河动作的数学模型](#)

[6.3 搜索算法](#)

[6.3.1 状态树的遍历](#)

[6.3.2 剪枝和重复状态判断](#)

[6.4 算法实现](#)

[6.5 总结](#)

[6.6 参考资料](#)

[第7章 稳定匹配与舞伴问题](#)

[7.1 稳定匹配问题](#)

[7.1.1 什么是稳定匹配](#)

[7.1.2 Gale-Shapley算法原理](#)

[7.2 Gale-Shapley算法的应用实例](#)

[7.2.1 算法实现](#)

[7.2.2 改进优化：空间换时间](#)

[7.3 有多少稳定匹配](#)

[7.3.1 穷举所有的完美匹配](#)

[7.3.2 不稳定因素的判断算法](#)

[7.3.3 穷举的结果](#)

[7.4 二部图与二分匹配](#)

[7.4.1 最大匹配与匈牙利算法](#)

[7.4.2 带权匹配与Kuhn-Munkres算法](#)

[7.5 总结](#)

[7.6 参考资料](#)

[第 8 章 爱因斯坦的思考题](#)

[8.1 问题的答案](#)

[8.2 分析问题的数学模型](#)

[8.2.1 基本模型定义](#)

[8.2.2 线索模型定义](#)

[8.3 算法设计](#)

[8.3.1 穷举所有的组合结果](#)

[8.3.2 利用线索判定结果的正确性](#)

[8.4 总结](#)

[8.5 参考资料](#)

[第 9 章 项目管理与图的拓扑排序](#)

[9.1 AOV网和AOE网](#)

[9.2 拓扑排序](#)

[9.2.1 拓扑排序的基本过程](#)

[9.2.2 按照活动开始时间排序](#)

[9.3 关键路径算法](#)

[9.3.1 什么是关键路径](#)

[9.3.2 计算关键路径的算法](#)

[9.4 总结](#)

[9.5 参考资料](#)

[第 10 章 RLE压缩算法与PCX图像文件格式](#)

[10.1 RLE压缩算法](#)

[10.1.1 连续重复数据的处理](#)

[10.1.2 连续非重复数据的处理](#)

[10.1.3 算法实现](#)

[10.2 RLE与PCX图像文件格式](#)

[10.2.1 PCX图像文件格式](#)

[10.2.2 PCX_RLE算法](#)

[10.2.3 256色PCX文件的解码和显示](#)

[10.3 总结](#)

[10.4 参考资料](#)

[第 11 章 算法与历法](#)

[11.1 格里历（公历）生成算法](#)

[11.1.1 格里历的历法规则](#)

[11.1.2 今天星期几](#)

[11.1.3 生成日历的算法](#)

[11.1.4 日历变更那点事儿](#)

[11.2 二十四节气的天文学计算](#)

[11.2.1 二十四节气的起源](#)

[11.2.2 二十四节气的天文学定义](#)

[11.2.3 VSOP-82/87行星理论](#)

[11.2.4 误差修正——章动](#)

[11.2.5 误差修正——光行差](#)

[11.2.6 用牛顿迭代法计算二十四节气](#)

[11.3 农历朔日（新月）的天文学计算](#)

[11.3.1 日月合朔的天文学定义](#)

- [11.3.2 ELP-2000/82月球理论](#)
- [11.3.3 误差修正——地球轨道离心率修正](#)
- [11.3.4 误差修正——黄经摄动](#)
- [11.3.5 月球地心视黄经和最后的修正——地球章动](#)
- [11.3.6 用牛顿迭代法计算日月合朔](#)
- [11.4 农历的生成算法](#)
 - [11.4.1 中国农历的起源与历法规则](#)
 - [11.4.2 中国农历的推算](#)
 - [11.4.3 一个简单的“年历”](#)
- [11.5 总结](#)
- [11.6 参考资料](#)
- [第 12 章 实验数据与曲线拟合](#)
 - [12.1 曲线拟合](#)
 - [12.1.1 曲线拟合的定义](#)
 - [12.1.2 简单线性数据拟合的例子](#)
 - [12.2 最小二乘法曲线拟合](#)
 - [12.2.1 最小二乘法原理](#)
 - [12.2.2 高斯消元法求解方程组](#)
 - [12.2.3 最小二乘法解决“速度与加速度”实验](#)
 - [12.3 三次样条曲线拟合](#)
 - [12.3.1 插值函数](#)
 - [12.3.2 样条函数的定义](#)
 - [12.3.3 边界条件](#)
 - [12.3.4 推导三次样条函数](#)
 - [12.3.5 追赶法求解方程组](#)
 - [12.3.6 三次样条曲线拟合算法实现](#)
 - [12.3.7 三次样条曲线拟合的效果](#)
 - [12.4 总结](#)
 - [12.5 参考资料](#)
- [第 13 章 非线性方程与牛顿迭代法](#)
 - [13.1 非线性方程求解的常用方法](#)
 - [13.1.1 公式法](#)
 - [13.1.2 二分逼近法](#)
 - [13.2 牛顿迭代法的数学原理](#)
 - [13.3 用牛顿迭代法求解非线性方程的实例](#)
 - [13.3.1 导函数的求解与近似公式](#)
 - [13.3.2 算法实现](#)
 - [13.4 参考资料](#)
- [第 14 章 计算几何与计算机图形学](#)
 - [14.1 计算几何的基本算法](#)
 - [14.1.1 点与矩形的关系](#)
 - [14.1.2 点与圆的关系](#)
 - [14.1.3 矢量的基础知识](#)
 - [14.1.4 点与直线的关系](#)
 - [14.1.5 直线与直线的关系](#)
 - [14.1.6 点与多边形的关系](#)
 - [14.2 直线生成算法](#)

[14.2.1 什么是光栅图形扫描转换](#)

[14.2.2 数值微分法](#)

[14.2.3 Bresenham算法](#)

[14.2.4 对称直线生成算法](#)

[14.2.5 两步算法](#)

[14.2.6 其他直线生成算法](#)

[14.3 圆生成算法](#)

[14.3.1 圆的八分对称性](#)

[14.3.2 中点画圆法](#)

[14.3.3 改进的中点画圆法——Bresenham算法](#)

[14.3.4 正负判定画圆法](#)

[14.4 椭圆生成算法](#)

[14.4.1 中点画椭圆法](#)

[14.4.2 Bresenham椭圆算法](#)

[14.5 多边形区域填充算法](#)

[14.5.1 种子填充算法](#)

[14.5.2 扫描线填充算法](#)

[14.5.3 改进的扫描线填充算法](#)

[14.5.4 边界标志填充算法](#)

[14.6 总结](#)

[14.7 参考资料](#)

第15章 音频频谱和均衡器与傅里叶变换算法

[15.1 实时频谱显示的原理](#)

[15.2 离散傅里叶变换](#)

[15.2.1 什么是傅里叶变换](#)

[15.2.2 傅里叶变换原理](#)

[15.2.3 快速傅里叶变换算法的实现](#)

[15.3 傅里叶变换与音频播放的实时频谱显示](#)

[15.3.1 频域数值的特点分析](#)

[15.3.2 从音频数据到功率频谱](#)

[15.3.3 音频播放时实时频谱显示的例子](#)

[15.4 破解电话号码的小把戏](#)

[15.4.1 拨号音的频谱分析](#)

[15.4.2 根据频谱数据反推电话号码](#)

[15.5 离散傅里叶逆变换](#)

[15.5.1 快速傅里叶逆变换的推导](#)

[15.5.2 快速傅里叶逆变换的算法实现](#)

[15.6 利用傅里叶变换实现频域均衡器](#)

[15.6.1 频域均衡器的实现原理](#)

[15.6.2 频域信号的增益与衰减](#)

[15.6.3 均衡器的实现——仿Foobar的18段均衡器](#)

[15.7 总结](#)

[15.8 参考资料](#)

第16章 全局最优解与遗传算法

[16.1 遗传算法的原理](#)

[16.1.1 遗传算法的基本概念](#)

[16.1.2 遗传算法的处理流程](#)

[16.2 遗传算法求解0-1背包问题](#)

[16.2.1 基因编码和种群初始化](#)

[16.2.2 适应度函数](#)

[16.2.3 选择算子设计与轮盘赌算法](#)

[16.2.4 交叉算子设计](#)

[16.2.5 变异算子设计](#)

[16.2.6 这就是遗传算法](#)

[16.3 总结](#)

[16.4 参考资料](#)

[第17章 计算器程序与大整数计算](#)

[17.1 哦，溢出了，出洋相的计算器程序](#)

[17.2 大整数计算的原理](#)

[17.2.1 大整数加法](#)

[17.2.2 大整数减法](#)

[17.2.3 大整数乘法](#)

[17.2.4 大整数除法与模](#)

[17.2.5 大整数乘方运算](#)

[17.3 大整数类的使用](#)

[17.3.1 与Windows的计算器程序一决高下](#)

[17.3.2 最大公约数和最小公倍数](#)

[17.3.3 用扩展欧几里得算法求模的逆元](#)

[17.4 总结](#)

[17.5 参考资料](#)

[第18章 RSA算法——加密与签名](#)

[18.1 RSA算法的开胃菜](#)

[18.1.1 将模幂运算转化为模乘运算](#)

[18.1.2 模乘运算与蒙哥马利算法](#)

[18.1.3 模幂算法](#)

[18.1.4 素数检验与米勒-拉宾算法](#)

[18.2 RSA算法原理](#)

[18.2.1 RSA算法的数学理论](#)

[18.2.2 加密和解密算法](#)

[18.2.3 RSA算法的安全性](#)

[18.3 数据块分组加密](#)

[18.3.1 字节流与大整数的转换](#)

[18.3.2 PKCS与OAEP加密填充模式](#)

[18.3.3 数据加密算法实现](#)

[18.3.4 数据解密算法实现](#)

[18.4 RSA签名与身份验证](#)

[18.4.1 RSASSA-PKCS与RSASSA-PSS签名填充模式](#)

[18.4.2 签名算法实现](#)

[18.4.3 验证签名算法实现](#)

[18.5 总结](#)

[18.6 参考资料](#)

[第19章 数独游戏](#)

[19.1 数独游戏的规则与技巧](#)

[19.1.1 数独游戏的规则](#)

[19.1.2 数独游戏的常用技巧](#)

[19.2 计算机求解数独问题](#)

[19.2.1 建立问题的数学模型](#)

[19.2.2 算法实现](#)

[19.2.3 与传统穷举方法的结果对比](#)

[19.3 关于数独的趣味话题](#)

[19.3.1 数独游戏有多少终盘](#)

[19.3.2 史上最难的数独游戏](#)

[19.4 总结](#)

[19.5 参考资料](#)

[第 20 章 华容道游戏](#)

[20.1 华容道游戏介绍](#)

[20.2 自动求解的算法原理](#)

[20.2.1 定义棋盘的局面](#)

[20.2.2 算法思路](#)

[20.3 自动求解的算法实现](#)

[20.3.1 棋局状态与Zobrist哈希算法](#)

[20.3.2 重复棋局和左右镜像的处理](#)

[20.3.3 正确结果的判断条件](#)

[20.3.4 武将棋子的移动](#)

[20.3.5 棋局的搜索算法](#)

[20.4 总结](#)

[20.5 参考资料](#)

[第 21 章 A*寻径算法](#)

[21.1 寻径算法演示程序](#)

[21.2 Dijkstra算法](#)

[21.2.1 Dijkstra算法原理](#)

[21.2.2 Dijkstra算法实现](#)

[21.2.3 Dijkstra算法演示程序](#)

[21.3 带启发的搜索算法——A*算法](#)

[21.3.1 A*算法原理](#)

[21.3.2 常用的距离评估函数](#)

[21.3.3 A*算法实现](#)

[21.4 总结](#)

[21.5 参考资料](#)

[第 22 章 俄罗斯方块游戏](#)

[22.1 俄罗斯方块游戏规则](#)

[22.2 俄罗斯方块游戏人工智能的算法原理](#)

[22.2.1 影响评价结果的因素](#)

[22.2.2 常用的俄罗斯方块游戏人工智能算法](#)

[22.2.3 Pierre Dellacherie评估算法](#)

[22.3 Pierre Dellacherie算法实现](#)

[22.3.1 基本数学模型和数据结构定义](#)

[22.3.2 算法实现](#)

[22.4 总结](#)

[22.5 参考资料](#)

[第 23 章 博弈树与棋类游戏](#)

- [23.1 棋类游戏的AI](#)
 - [23.1.1 博弈与博弈树](#)
 - [23.1.2 极大极小值搜索算法](#)
 - [23.1.3 负极大极搜索算法](#)
 - [23.1.4 “ \$\alpha\$ - \$\beta\$ ”剪枝算法](#)
 - [23.1.5 估值函数](#)
 - [23.1.6 置换表与哈希函数](#)
 - [23.1.7 开局库与终局库](#)
- [23.2 井字棋——最简单的博弈游戏](#)
 - [23.2.1 棋盘与棋子的数学模型](#)
 - [23.2.2 估值函数与估值算法](#)
 - [23.2.3 如何产生走法（落子方法）](#)
- [23.3 奥赛罗棋（黑白棋）](#)
 - [23.3.1 棋盘与棋子的数学模型](#)
 - [23.3.2 估值函数与估值算法](#)
 - [23.3.3 搜索算法实现](#)
 - [23.3.4 最终结果](#)
- [23.4 五子棋](#)
 - [23.4.1 棋盘与棋子的数学模型](#)
 - [23.4.2 估值函数与估值算法](#)
 - [23.4.3 搜索算法实现](#)
 - [23.4.4 最终结果](#)
- [23.5 总结](#)
- [23.6 参考资料](#)

[附录A 算法设计的常用技巧](#)

- [A.1 数组下标处理](#)
- [A.2 一重循环实现两重循环的功能](#)
- [A.3 棋盘（迷宫）类算法方向遍历](#)
- [A.4 代码的一致性处理技巧](#)
- [A.5 链表和数组的配合使用](#)
- [A.6 “以空间换时间”的常用技巧](#)
- [A.7 利用表驱动避免长长的switch-case](#)

[附录B 一个棋类游戏的设计框架](#)

- [B.1 代码框架的整体结构](#)
- [B.2 代码框架的使用方法](#)

序一

读《算法的乐趣》的乐趣超出了我的预料。

说到算法，大部分计算机专业的同学的第一反应估计是MIT出版社的经典教材《算法导论》（*Introduction to Algorithms*）。这是一本由浅入深的好书，堪称“神书”——别看书挺厚，但是对初学者来说很难看懂的问题也娓娓道来，让人看一遍就明白；而且作者用最简单的英语词汇和句法写书，以至于世界各地的学生们，不需要英语很好，即可读懂原版。只是看完这本大部头之后，总有一些意犹未尽的感觉——对我们日常生活中常见的比如音乐播放器里以及电子游戏里的算法并没有太多介绍。而这些正是《算法的乐趣》中主要的部分。

在Amazon上，另外两本排名靠前的经典算法教材是Jon Kleinberg的《算法设计》（*Algorithm Design*）和Steven S. Skiena的《算法设计手册》（*The Algorithm Design Manual*）。这两本出自名家之手的教材和很多教材一样，按照算法的类型或者背后的设计思路来组织内容。这是教材应该做的，“授人以鱼不如授人以渔”，传授思路而不是算法本身是教材的写作目的。可是算法最有意思的地方首先在于算法本身，因为算法是为了解决实际问题而设计的，所以让大家认识到算法奥妙的自然顺序应该是先展示有趣的问题，再展示优雅的算法，最后归纳设计思路。而这正是《算法的乐趣》吸引人的地方。

说到乐趣，总让我想起我学习和使用数学知识的经历。虽然我的学位是关于统计机器学习的，而且毕业后一直从事相关工作，但是我从小学一年级到博士第三年都对数学毫无兴趣，因为学校的老师和数学成绩好的同学都说不明白数学的用处，以至于我一直以为数学的作用只是锻炼和展示自己的聪明，博得老师的表扬，成为陈景润那样为国争光的英雄。而这些对我实在没有吸引力，而且我认为恐怕对绝大部分学生都没什么吸引力。

我认识到数学的价值，是因为在博士第三年把研究方向换到了统计机器学习。在读教材的时候，我曾想验证“数学无用”，所以费尽心力地试图写一个程序来判断一个64×64像素的图片里到底是数字“1”还是数字“9”，却发现无论如何也很难写一个有效的程序；可是利用教材里的数学知识却能设计和“训练”一个数学模型，准确地识别任意字符。因为体会到了数学的用处，我兴奋地用了一年的时间复习大学本科的数学课程，然后才读懂了人工智能的专业教材和论文。此后才有所创新，发表论文，到博士毕业。这整个过程用了三年，而效果超过了之前19年数学教育的效果。

在这个过程中，我自然而然地开始注意数学知识的前因（比如为什么人们会关注长度、面积，怎么会有人考虑勾股定理这样的规律）以及后果（今天的数学知识能给物理学和机器智能带来什么样的帮助），也开始归纳和了解各种数学系统背后的规律，能体会哥德尔定理阐述的意思。当然，也破除了“数学是各种科学之母”之类的迷信，数学当然不是“科学之母”，而是“科学之子”，是先有物理学、力学和天文学，才有的数学；先有应用场景后有工具，先有探索后有归纳。

算法也是如此。先有工程问题需要解决，算法是解法，设计算法是寻求解法。虽然算法作为一门科学是归纳寻求解法的思路，但学习这种归纳法的前提是能体会各种具体算法的用处和效果。意识到这一点，自然也就破除了诸如“学好数学才能学好算法”之类的迷信。而把算法解决的各种有趣问题罗列出来，把算法的可爱之处展示给愿意发现和体会生活中点滴乐趣的读者们，正是《算法的乐趣》在技术价值之外的一层社会价值。

十年前，当我们坐在课堂里学习算法的时候，我们学到的是如何用人脑寻求解法，然后把解法写成程序，让计算机照着执行去解决问题。这是“经典算法”。最近十几年，随着Internet产业的兴起，Internet服务在不断取代原来由人提供的服务，这就要求机器拥有一定程度上能取代人的“智能”。在搜索引擎、推荐系统和广告系统等各个领域里，类似上述“识别数字”的问题越来越多，而人工智能和机器学习的应用也越来越深入我们的生活。机器学习算法的设计目标和“经典算法”不同——不是让人来想解法，而是让计算机从数据归纳知识——有了这些知识，计算机就能自己寻求解法。

虽然经典算法和机器学习算法之间的差别大得如同一场革命，但是由经典而入机器学习的过程却是自然而然的。比如《算法的乐趣》中介绍的曲线拟合问题，就是supervised learning（有监督学习），而音乐播放器里常用的傅里叶变换和其他时域频域变换则是unsupervised learning（无监督学习）的技术基础，棋类游戏算法是博弈论和reinforcement learning（强化学习）的经典例子。我常见有朋友从读数学教材开始探索机器学习和人工智能算法，也常看到有人不堪忍受长时间缺乏乐趣的探索以至于半途而废。如果是这样，也许不如从《算法的乐趣》开始这个探索过程。

我曾经以为从乐趣出发阐述算法的书会从西方发芽，没想到先看到了一本中文书。这真超出了我的预料。

王益

LinkedIn高级主任分析师

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《算法的乐趣（图灵原创）》王晓华 著.epub

请登录 <https://shgis.cn/post/323.html> 下载完整文档。

手机端请扫码查看：

