



【韩】具宗万 著 崔盛一 译

算法问题 实战策略



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



【韩】具宗万 著 崔盛一 译

算法问题 实战策略

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

算法问题实战策略 / (韩) 具宗万著 ; 崔盛一译.
— 北京 : 人民邮电出版社, 2015.3
(图灵程序设计丛书)
ISBN 978-7-115-38462-1

I. ①算… II. ①具… ②崔… III. ①计算机算法
IV. ①TP301.6

中国版本图书馆CIP数据核字 (2015) 第017771号

版 权 声 明

프로그래밍 대회에서 배우는 알고리즘 문제 해결 전략 © 2012 by JongMan Koo
All rights reserved.
Translation rights arranged by INSIGHT
through Shinwon Agency Co., Korea
Simplified Chinese Translation Copyright © 2015 by Posts & Telecom Press.

本书中文简体字版由 Insight 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

内 容 提 要

本书收录程序设计竞赛经典试题, 在解题过程中讲解各种算法设计技巧和数据结构, 培养读者的解题能力。读者可亲自编写各章习题程序并获得评分, 所有示例均附有解题过程及详细说明。

-
- ◆ 著 [韩] 具宗万
 - 译 崔盛一
 - 责任编辑 傅志红
 - 执行编辑 陈 曦
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 46.75
 - 字数: 1104千字 2015年3月第1版
 - 印数: 1-4 000册 2015年3月河北第1次印刷
 - 著作权合同登记号 图字: 01-2013-8254 号
-

定价: 119.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

作 者 序

从我决定写书至今，已经过了 6 年。虽然拙作尚有许多部分需要完善，但我认为，能够让读者读到一本略有欠缺的书，总比一本永远未能出版的完美的书要好得多。基于这种想法，我鼓足勇气出版了本书。

我攻读硕士时未曾专门研究过算法，实际经验也不是很丰富。因此，一想到自己要编写一本解决算法问题这种难解主题的大部头书籍，就觉得是不是想法有些轻率。不过，我想把学习编程和解决问题过程中得到的成就感与喜悦与他人共享，最终还是下定了决心。希望本书能让更多人再次感受到编程的快乐。

各位从我写书耗费的时间就可知道，编写本书真是历经千辛万苦。无数次删除重写，不知如何继续的时候，我曾多次想过，究竟能否完成本书。所幸有许多比我更优秀的专家给予的支持、帮助和鼓励，本书最终还是面世了。真想对帮助过我的人一一表示感谢，不过为了赶在年内出版，此处只列出他们的姓名。

高锡敏、权洙玄、金光德、金光燮、金东健、金东赫、金敏慧、金字现、金融永、金真虎、金灿民、金太勋、金孝胜、柳元河、柳贤钟、文成益、朴贤庾、宋英俊、申海秀、安圣远、吴时英、刘元石、尹形锡、李涛京、李昌现、李太尹、李现钟、李候妍、李熙胜、郑瑟雅、郑珠英、郑贤焕、赵淳姬、崔琪猷、崔伯俊、崔盛妍、崔余敏、崔钟旭、何载承、何载勋、黄意权、黄仁旭，感谢各位。他们打造了本书中的亮点，而失误和错误都是本人能力不够造成的。

感谢一直支持我的 algospot 运营团队。我初次接触程序设计竞赛时，延世大学 YUTAR 信息特技社团的前辈们和母校的教授们教会我许多技巧，谢谢大家。感谢当初间接提醒我写下本书的金昌俊先生，以及 Insight 出版社韩基成社长和金承虎编辑，他们在原计划 6 个月而最终延长了近十二倍的写作时间中一直耐心等待，谢谢。

最后，我要把本书献给亲爱的妻子和儿子，即使我用写作当借口逃避家务，他们也始终体谅我。

具宗万

2012 年 10 月，芝加哥

目 录

第一部分 开始解决问题

第 1 章 解决问题与程序设计竞赛	4
1.1 引言	4
1.2 程序设计竞赛	4
1.3 阅读本书的方法	7
1.4 值得参加的程序设计竞赛	8
1.5 对赛前准备工作的一些建议	9
1.6 续读	12
第 2 章 解决问题概述	13
2.1 引言	13
2.2 解决问题的过程	13
2.3 解决问题的策略	17
2.4 续读	26
第 3 章 编码与调试	27
3.1 引言：不要忽视编码的重要性	27
3.2 编写优秀代码的原则	27
3.3 常见失误	32
3.4 调试与测试	39
3.5 变量的取值范围	42
3.6 理解实数型数据类型	46
3.7 续读	55

第二部分 算法分析

第 4 章 分析算法的时间复杂度	60
4.1 引言	60
4.2 线性时间算法	62
4.3 次线性时间算法	65

4.4 指数时间算法	67
4.5 时间复杂度	70
4.6 推测执行时间	76
4.7 计算复杂度类：P、NP、NP-完备	81
4.8 续读	84
第 5 章 算法正确性证明	85
5.1 引言	85
5.2 数学归纳法和循环不变式	86
5.3 归谬法	90
5.4 其他技巧	92
5.5 续读	95

第三部分 算法设计范式

第 6 章 暴力解决法	99
6.1 引言	99
6.2 递归调用和穷举搜索法	100
6.3 练习题：郊游（习题 ID：PICNIC，难度：低）	106
6.4 解题：郊游	107
6.5 练习题：盖游戏板（习题 ID：BOARDCOVER，难度：低）	109
6.6 解题：盖游戏板	111
6.7 优化问题	113
6.8 练习题：时钟同步（习题 ID：CLOCKSYNC，难度：中）	116
6.9 解题：时钟同步	117
6.10 常见穷举搜索类型	119
第 7 章 分治法	120
7.1 引言	120

7.2 练习题：四叉树问题（题目 ID： QUADTREE，难度：低）	130
7.3 解题：四叉树问题	131
7.4 练习题：切割篱笆（习题 ID： FENCE，难度：中）	134
7.5 解题：切割篱笆	135
7.6 练习题：粉丝见面会（题目 ID： FANMEETING，难度：高）	139
7.7 解题：粉丝见面会	141
第 8 章 动态规划法	143
8.1 引言	143
8.2 练习题：通配符（习题 ID： WILDCARD，难度：中）	151
8.3 解题：通配符	152
8.4 典型优化问题	156
8.5 练习题：合并 LIS（题目 ID： JLIS，难度：低）	163
8.6 解题：合并 LIS	164
8.7 练习题：背诵圆周率（题目 ID： PI，难度：低）	166
8.8 解题：背诵圆周率	167
8.9 练习题：Quantization（题目 ID： QUANTIZE，难度：中）	169
8.10 解题：Quantization	170
8.11 所有可能的个数与概率	174
8.12 练习题：非对称铺设（题目 ID： ASYMTILING，难度：低）	180
8.13 解题：非对称铺设	181
8.14 练习题：多联骨牌（题目 ID： POLY，难度：中）	183
8.15 解题：多联骨牌	185
8.16 练习题：逃狱的韩尼拔博士（题目 ID： NUMB3RS，难度：中）	187
8.17 解题：逃狱的韩尼拔博士	189
第 9 章 动态规划技巧	194
9.1 计算优化问题的实际答案	194
9.2 练习题：打包行李（题目 ID： PACKING，难度：中）	195
9.3 解题：打包行李	197
9.4 练习题：光学字符识别（题目 ID： OCR，难度：高）	199
9.5 解题：光学字符识别	201
9.6 计算第 k 个答案	204
9.7 练习题：第 k 个最大递增子序列 （题目 ID：KLIS，难度：高）	209
9.8 解题：第 k 个最长递增子序列	210
9.9 练习题：龙曲线（题目 ID： DRAGON，难度：中）	214
9.10 解题：龙曲线	216
9.11 对非整数型输入的制表	219
9.12 练习题：韦布巴津（题目 ID： ZIMBABWE，难度：高）	224
9.13 解题：韦布巴津	225
9.14 练习题：恢复实验数据（题目 ID： RESTORE，难度：中）	230
9.15 解题：恢复实验数据	231
9.16 组合游戏	234
9.17 练习题：数字游戏（题目 ID： NUMBERGAME，难度：低）	239
9.18 解题：数字游戏	240
9.19 练习题：方块游戏（题目 ID： BLOCKGAME，难度：中）	242
9.20 解题：方块游戏	243
9.21 迭代动态规划法	245
9.22 练习题：回转寿司（题目 ID： SUSHI，难度：中）	249
9.23 解题：回转寿司	250
9.24 练习题：Genius（题目 ID： GENIUS，难度：中）	253
9.25 解题：Genius	254
9.26 续读	256
第 10 章 贪心法	257
10.1 引言	257
10.2 练习题：加热便当（题目 ID： LUNCHBOX，难度：低）	264
10.3 解题：加热便当	265

10.4 练习题: 合并字符串 (题目 ID: STRJOIN, 难度: 中)	268	13.3 练习题: 提高获胜率 (题目 ID: RATIO, 难度: 低)	338
10.5 解题: 合并字符串	269	13.4 解题: 提高获胜率	339
10.6 练习题: 米那斯雅诺 (题目 ID: MINASTIRITH, 难度: 高)	273	13.5 三叉搜索	341
10.7 解题: 米那斯雅诺	275	13.6 练习题: 花粉化石 (题目 ID: FOSSIL, 难度: 高)	346
第 11 章 组合搜索	281	13.7 解题: 花粉化石	347
11.1 引言	281	13.8 其他主题	351
11.2 组合搜索的方法	283	第 14 章 整数论	352
11.3 练习题: 盖游戏板2 (题目 ID: BOARDCOVER2, 难度: 低)	298	14.1 引言	352
11.4 解题: 盖游戏板2	299	14.2 素数	352
11.5 练习题: 患有严重过敏症的朋 友们 (题目 ID: ALLERGY, 难度: 中)	303	14.3 练习题: 密码486 (题目 ID: PASS486, 难度: 中)	357
11.6 解题: 患有严重过敏症的朋友们	304	14.4 解题: 密码486	357
11.7 练习题: 数谜 (题目 ID: KAKURO2, 难度: 中)	307	14.5 欧几里得算法	360
11.8 解题: 数谜	309	14.6 练习题: 魔法药水 (题目 ID: POTION, 难度: 中)	361
11.9 续读	315	14.7 解题: 魔法药水	362
第 12 章 将优化问题转换为决策 问题求解	316	14.8 模运算	364
12.1 引言	316	14.9 续读	366
12.2 练习题: 南极基地 (题目 ID: ARCTIC, 难度: 低)	320	第 15 章 计算几何	367
12.3 解题: 南极基地	321	15.1 引言	367
12.4 练习题: 加拿大旅行 (题目 ID: CANADATRIP, 难度: 中)	323	15.2 计算几何的工具	367
12.5 解题: 加拿大旅行	324	15.3 相交、距离、面积	373
12.6 练习题: 退选课程 (题目 ID: WITHDRAWAL, 难度: 高)	326	15.4 练习题: 弹球模拟 (题目 ID: PINBALL, 难度: 高)	377
12.7 解题: 退选课程	327	15.5 解题: 弹球模拟	379
第四部分 一些著名的算法		15.6 多边形	383
第 13 章 数值分析	331	15.7 练习题: 金银岛 (题目 ID: TREASURE, 难度: 高)	386
13.1 引言	331	15.8 解题: 金银岛	387
13.2 二分法	331	15.9 练习题: 是傻子? 不是傻子? (题目 ID: NERDS, 难度: 中)	390
		15.10 解题: 是傻子? 不是傻子?	392
		15.11 计算几何算法设计范式	396
		15.12 常见失误与注意事项	403
		15.13 续读	404

第五部分 基本数据结构

第 16 章 位掩码	410
16.1 引言	410
16.2 利用位掩码实现集合	413
16.3 位掩码应用示例	417
16.4 练习题: 毕业学期 (题目 ID: GRADUATION, 难度: 中)	420
16.5 解题: 毕业学期	422
16.6 续读	424
第 17 章 部分和	425
17.1 引言	425
17.2 练习题: 圣诞娃娃 (题目 ID: CHRISTMAS, 难度: 中)	429
17.3 解题: 圣诞娃娃	430
17.4 其他学习内容	432
第 18 章 线性数据结构	433
18.1 引言	433
18.2 动态数组	433
18.3 链表	437
18.4 动态数组和链表的比较	440
18.5 练习题: 约瑟夫斯 (题目 ID: JOSEPHUS, 难度: 低)	440
18.6 解题: 约瑟夫斯	441
18.7 续读	442
第 19 章 队列、栈以及双端队列	443
19.1 引言	443
19.2 队列、栈以及双端队列的实现方法	444
19.3 队列与栈的应用	445
19.4 练习题: 不匹配括号 (题目 ID: BRACKETS2, 难度: 低)	448
19.5 解题: 不匹配括号	449
19.6 练习题: 分析外星信号 (题目 ID: ITES, 难度: 中)	450
19.7 解题: 分析外星信号	451
第 20 章 字符串	455
20.1 引言	455

20.2 字符串检索	456
20.3 练习题: 宰河的保险箱 (题目 ID: JAEHASAFE, 难度: 中)	466
20.4 解题: 宰河的保险箱	467
20.5 后缀数组	468
20.6 练习题: 口头禅 (题目 ID: HABIT, 难度: 中)	476
20.7 解题: 口头禅	477
20.8 续读	478

第六部分 树

第 21 章 树的实现与遍历	481
21.1 引言	481
21.2 树的遍历	483
21.3 练习题: 变更树的遍历顺序 (题目 ID: TRAVERSAL, 难度: 低)	484
21.4 解题: 变更树的遍历顺序	486
21.5 练习题: 要塞 (题目 ID: FORTRESS, 难度: 中)	487
21.6 解题: 要塞	488
第 22 章 二叉搜索树	493
22.1 引言	493
22.2 二叉搜索树的定义和操作	493
22.3 时间复杂度分析与平衡二叉搜索树	496
22.4 练习题: 是呆子? 不是呆子? 2 (题目 ID: NERD2, 难度: 中)	496
22.5 解题: 是呆子? 不是呆子? 2	498
22.6 直接实现平衡二叉搜索树: 树堆	501
22.7 练习题: 反转插入排序 (题目 ID: INSERTION, 难度: 中)	508
22.8 解题: 反转插入排序	509
第 23 章 优先级队列和堆	511
23.1 引言	511
23.2 堆的定义与实现方法	512
23.3 练习题: 变化的中间值 (题目 ID: RUNNINGMEDIAN, 难度: 低)	518
23.4 解题: 变化的中间值	519

第 24 章	区间树	521
24.1	区间树：区间相关问题解答	521
24.2	练习题：登山路（题目 ID： MORDDR，难度：中）	527
24.3	解题：登山路	528
24.4	练习题：寻根问祖（题目 ID： FAMILYTREE，难度：高）	529
24.5	解题：寻根问祖	530
24.6	树状数组：快速而简单的区间和	533
24.7	练习题：计算插入排序的时间 （题目 ID：MEASURETIME， 难度：中）	536
24.8	解题：计算插入排序的时间	537

第 25 章 互斥集合	541
25.1 引言	541
25.2 练习题: 编辑器之争 (题目 ID: EDITORWARS, 难度: 中)	546
25.3 解题: 编辑器之争	548

第 26 章	字典树	553
26.1	引言	553
26.2	练习题: 再见, 谢谢所有的鱼 (题目 ID: SOLONG, 难度: 中)	557
26.3	解题: 再见, 谢谢所有的鱼	559
26.4	利用字典树检索多重字符串	563
26.5	练习题: 安全终结者 (题目 ID: NH, 难度: 高)	569
26.6	解题: 安全终结者	570

第七部分 图

第 27 章 图的表示方式及定义	576
27.1 引言	576
27.2 图的应用示例	579
27.3 隐式图结构	580
27.4 图的几种表示法	581

第 28 章 图的深度优先搜索 585

28.1 引言 585

28.2	练习题: 古语词典 (习题 ID: DICTIONARY, 难度: 低)	590
28.3	解题: 古语词典	591
28.4	欧拉回路	594
28.5	练习题: 有限单词接龙 (题目 ID: WORDCHAIN, 难度: 低)	597
28.6	解题: 有限单词接龙	598
28.7	理论背景及应用	602
28.8	练习题: 安装监控摄像头 (题目 ID: GALLERY, 难度: 中)	613
28.9	解题: 安装监控摄像头	614
28.10	练习题: 安排会议室 (题目 ID: MEETINGROOM, 难度: 高)	616
28.11	解题: 安排会议室	618

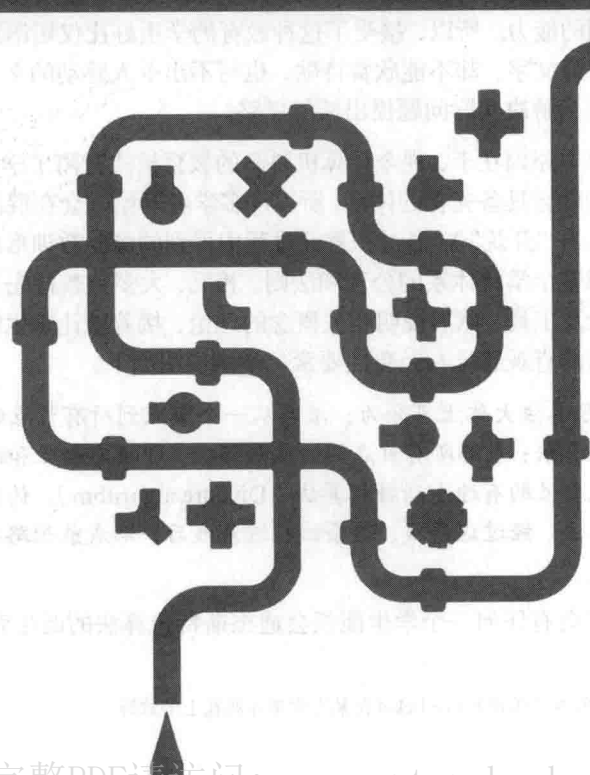
第 29 章 图的宽度优先搜索	625
29.1 引言	625
29.2 练习题: 排序游戏 (题目 ID: SORTGAME, 难度: 中)	629
29.3 解题: 排序游戏	630
29.4 练习题: 儿童节 (题目 ID: CHILDRENDAY, 难度: 高)	633
29.5 解题: 儿童节	634
29.6 最短路径策略	637
29.7 练习题: 汉诺塔 (题目 ID: HANOI4B, 难度: 中)	648
29.8 解题: 汉诺塔	650

第 30 章	最短路径问题	653
30.1	引言	653
30.2	迪杰斯特拉最短路径算法	654
30.3	练习题: 信号路由 (题目 ID: ROUTING, 难度: 低)	661
30.4	解题: 信号路由	662
30.5	练习题: 消防车 (题目 ID: FIRETRUCKS, 难度: 中)	663
30.6	解题: 消防车	664
30.7	练习题: 铁人 N 项比赛 (题目 ID: NTHLON, 难度: 高)	665
30.8	解题: 铁人 N 项比赛	667

30.9 贝尔曼—福特最短路径算法	669	31.7 解题：选定旅行路线	700
30.10 练习题：时间旅行（题目 ID： TIMETRIP，难度：中）	674	第 32 章 网络流	705
30.11 解题：时间旅行	675	32.1 引言	705
30.12 弗洛伊德多源最短路径算法	677	32.2 福特—富尔克森算法	706
30.13 练习题：检查酒驾（题目 ID： DRUNKEN，难度：中）	682	32.3 网络建模	713
30.14 解题：检查酒驾	684	32.4 练习题：操纵比赛（题目 ID： MATCHFIX，难度：中）	715
30.15 练习题：竞选承诺（题目 ID： PROMISES，难度：中）	685	32.5 解题：操纵比赛	717
30.16 解题：竞选承诺	687	32.6 练习题：国家项目（题目 ID： PROJECTS，难度：高）	719
第 31 章 最小生成树	689	32.7 解题：国家项目	720
31.1 引言	689	32.8 二分图匹配	723
31.2 克鲁斯卡尔最小生成树算法	690	32.9 练习题：象（题目 ID：BISHOPS， 难度：中）	729
31.3 普里姆最小生成树算法	694	32.10 解题：象	730
31.4 练习题：局域网（题目 ID：LAN， 难度：低）	697	32.11 练习题：设置陷阱（题目 ID： TRAPCARD，难度：高）	732
31.5 解题：局域网	698	32.12 解题：设置陷阱	734
31.6 练习题：选定旅行路线（题目 ID： TPATH，难度：高）	699	32.13 其他学习内容	737

第一部分

开始解决问题



概述

程序设计是一门高难度的学科，很多接受过良好的计算机教育的学生在毕业后仍对程序设计感到吃力。不少软件公司常常怨声不断，招来的新人都毕业于名校计算机专业，却连一行程序代码都写不出来。即使这并不是程序设计的问题，而是大学教育的痼疾，但像程序设计这样能够真实体现“天才与庸才的效率相差 20 倍”^①这句话的行业并不多见。

这种现象的根本原因在于，大部分计算机科学相关的教育课程只教授程序设计的技术和知识，而忽视了自主应用的能力。所以，接受了这种教育的学生好比仅用语法书和词典学习外语的人一样，只能机械地解释文字，却不能欣赏诗歌，也写不出令人感动的文章。就算学习算法后可以解出练习题，却不能为实际问题提出新的思路。

造成这种悲剧的根本原因在于，现今计算机科学的教育模式背离了学科的实际发展方向。没有一个学科在发展之初就会具备完善的体系，所以许多学者开始时会在混沌中根据自己的直观感觉进行摸索（就是常说的“开荒”）。通过积累此过程中得到的经验教训形成学科基础，学者们会自然而然地创建出支撑整个学科体系的公理和法则。相反，大多数教材先介绍学科发展过程的最终结果——复杂的概念及工具，然后说明相关概念的理论，接着就让学生解答习题。这些教材完全排除了创建理论所需的直观感觉和经验性要素。举例说明如下。

“图的最短路径算法大体上可分为：求解从一个节点到所有节点的最短路径，即确定起点的最短路径算法；求解所有节点间的最短路径，即确定起点和终点的最短路径算法。前者中最具代表性的有迪杰斯特拉算法（Dijkstra algorithm）。伪代码如下所示。现在针对该图测试算法。经过这条线、这条线、这条线后，形成最短路径。好了，现在分析一下时间复杂度。”

在此堂课当中，不会有任何一个学生能领会迪杰斯特拉算法的诞生背景和设计时的关键思

^① 《Java 编程思想》等名著的作者 Bruce Eckel 在某大学毕业典礼上的致辞。

路。这种教育方式也许对学生系统化地了解学科发展的结论性知识有些帮助，但无法培养学生独立思考的能力。要想真正灵活运用所学知识，必须让学生们了解学科发展过程当中的重大发现和领悟。

了解这些发现和领悟的最佳途径就是程序设计竞赛。程序设计竞赛利用计算机科学领域中广泛使用的算法和数据结构，比拼参赛者解决考题及实现其算法的能力。竞赛提出的问题不仅需要利用已知的算法直接解答，还必须理解算法原理并加以变换，这对深入了解这些主题有很大的帮助。竞赛还会提供公正而快速的反馈信息，这也为自学提供了乐趣。

通过书中对程序设计竞赛题目的解答，读者能够直观地了解各种算法的设计技巧和数据结构，进而提高解决算法问题的能力。为此，书中详细记述了各种算法诞生的背景、缘由以及过程。习题由程序设计竞赛的真题组成，而且读者可以根据相应章节算法及其变换形式进行练习。本书的目标是，通过一些（较难）题目的解题过程，让读者体验各种概念及结构。

程序设计虽然很难，但同时也是一个很有趣的过程。解答问题亦是如此，解不出问题时的痛苦会被解题后的喜悦和充实感冲淡。希望各位能够愉快地阅读本书。

第 1 章

解决问题与程序设计竞赛

1.1 引言

当今世界，“程序员”这么一个职业名称可分为很多类型，比如 SI 程序员、数据库程序员、网页设计程序员、游戏设计程序员，等等。这些程序员在不同的开发环境下，利用不同的程序语言和工具编写程序，解决各自面临的问题。那么，要成为跨领域的优秀程序员需要具备什么样的条件呢？

程序设计就是解决问题

程序设计需要丰富的知识。可能程序员只是看起来在漫无目的地敲打键盘，其实他们脑海中对程序设计语言的特性、程序运行的硬件平台和操作系统、对所使用的库的注意事项等进行着综合性的思考。在满足程序运行可占用内存空间的要求和限定的运算时间内，尽可能编写简洁而可重复利用的代码。

了解这些限制条件和要求后找到最佳解决方法，这种能力就是成为跨领域程序员的必要条件，本书称这种能力为“解决问题的能力”。程序员掌握的程序语言、库、算法等方面的知识就好比一个个拼图块，“解决问题”就是将这些拼图块放入合适位置，最终完成整幅图案。

但解决问题的能力是十分抽象的，所以训练起来相当困难。想自学成才的程序员虽然可以一直学习新的程序语言和程序框架、开发方法，但无处可学如何对其进行整合。

那么，想成为优秀程序员有没有更好的途径呢？

1.2 程序设计竞赛

参加程序设计竞赛是培养解决问题能力的最佳途径。在目前各式各样的程序设计竞赛当中，

本书选取设计要求较少、有运算时间限制、比拼快速设计能力的竞赛。此处的设计要求不同于实际开发中的要求，而是以“读入某个值，然后计算出某个值”的形式给出。

下面针对未接触过程序设计竞赛的读者举个简单的示例。

题目：摇滚音乐节（难度：低，题目 ID：FESTIVAL）

要租用一个大的演出场地举办摇滚音乐节。这次音乐节将持续数日，每天有一支乐队举行演唱会。还未确定共邀请几个乐队，但已经与 L 个明星乐队交涉完毕，所以音乐节至少要举办 L 日。

要使用的演出场地每日租金各不相同，因此要妥当安排演出日程来节约成本。假设已经知道今后 N 日内的租金价格，在此期间连续租用 L 日，那么要使日均租金降至最低，应该怎样租呢？

假设未来六天的租金为 $\{3, 1, 2, 3, 1, 2\}$ ，已邀请了 3 支乐队，那么举办 4 天音乐节要比举办 3 天日均租金更低。因为举办 3 天的最低日均租金是从第二天开始租用到第四天，租金为 $(1+2+3)/3=2$ ；而从第二天开始一直租用到第五天的日均租金却只有 $(1+2+3+1)/4=7/4$ 。

运行时间及占用内存的限制条件

程序运算时间限定为 2 秒以内，占用内存不得超过 64MB。

输入

执行程序时，首先要求输入测试用例的个数 C ($C \leq 100$)，接下来的测试用例阶段第一行输入可租赁演出场地的最大天数 N ($1 \leq N \leq 1000$)，以及已邀请到的乐队数量 L ($1 \leq L \leq 1000$, $L \leq N$)，然后在第二行输入 N 个日租金数。租金数为小于 100 的自然数。

输出

对应各测试用例，每一行显示一个最小日均租金数，允许小于 10^{-7} 的绝对/相对误差。

输入示例

```
2
6 3
1 2 3 1 2 3
6 2
1 2 3 1 2 3
```

输出示例

```
1.7500000000
1.5000000000
```

从程序设计竞赛中学到的

程序设计竞赛的题目不同于实际开发过程中的规范，通过分析两者的不同，能得出竞赛题目

具有以下优势。

1. 程序设计竞赛中设计的程序绝对不会使用图形界面，只利用文本方式进行输入、输出等操作。少了这些累赘，参赛者就可以集中所有精力去解决问题。
2. 对运算时间和内存使用量有明确限制。在测评运行时，如果参赛者提交的程序超时或内存使用量超出限制，则无论计算结果是否正确，都将视为错误答案。程序设计竞赛的题目都是需要集中计算^①的，若不使用适当的算法和数据结构，就不可能设计出在限定时间内完成运算的程序。因此，参赛者需慎重选择要使用的算法，才能设计出满足限制条件的程序。对参赛者来说，程序设计竞赛是对丰富的算法设计技巧和数据结构进行实战的很好契机。竞赛提出的题目有些可以直接利用已知的算法解答，但还有些是必须在理解算法的原理后，再加以变换才能解答的题目。这对深刻理解其主题有着莫大的帮助。
3. 明确区分答案正确与否。实际开发中，通过单元测试或相互代码审查以测试代码的正确性。这种方法最多利用一两个输入的值测试程序代码。而程序设计竞赛利用多种多样的输入值测试运算的准确性，所以发生漏检错误的可能性较小。而且大部分的程序设计竞赛在赛后立刻公布解题正误，甚至在有些竞赛中，参赛者提交解题代码后就能当场得知结果。自己编写的代码能够得到快速、客观的意见反馈，这无疑是编写高质量程序的很好锻炼。
4. 提交程序后，运算时间和内存使用状况等信息会实时提供给参赛者。因此，可以一边修改自己的程序代码，一边观察修改对程序效率产生的影响。这种方式能够提高对逻辑结构复杂的程序进行性能预测的能力，对实际工作当中编写高性能程序有很大帮助。
5. 程序设计竞赛的程序规模都比较小，所以每个题目都要从头开始编写。这种特性使参赛者有机会认真研究开发大型项目时忽视的小细节。开发大项目时无法随意更改代码结构，有时还会为了整体性能而牺牲程序的优雅之感。但编写竞赛的小型代码时不用考虑这些问题，能够投入更多时间编写优雅的程序，成为编写出好代码的极佳练习。
6. 程序设计竞赛中，参赛者在多人竞争的环境下编写代码。从某种意义上讲，相互竞争是提高自身实力的一个动机。还值得注意的是，许多竞赛结束后会公开所有参赛者的源代码。实际开发中，不会有两个或两个以上的程序员同时编写相同功能的代码。但在竞赛中，所有参赛者均同时编写相同功能的代码。因此，参赛者有机会把自己编写的代码与世界顶尖程序员编写的代码进行比较，从而进一步提升自己的能力。

这些特点使得程序设计竞赛强调解决问题的能力，因此，竞赛中只保留必备要素，舍弃其他无关紧要的部分。所以，程序设计竞赛是学习解决问题的最佳环境。

各位是否也感到值得一试呢？

^① 程序运行时间主要耗费在利用 CPU 进行的计算上，而不是消耗在访问数据库、网络等操作上。

1.3 阅读本书的方法

本书结构

本书共由 7 个部分组成。为学好后半部分的内容，第一部分和第二部分主要介绍了一些背景知识，其中包括程序设计竞赛中解决问题的方法、编写代码和调试中的注意事项、算法的正确性证明以及效率分析等。不要以为此部分是基础性的内容而可以忽视，如果不能很好地理解这些基础知识，那么之后更深的内容就只能学到皮毛。本书的其他部分都将围绕着算法设计技巧和数据结构等程序设计竞赛解决问题的技术进行介绍。在这些部分将讲解各个算法诞生的背景以及设计过程，也包括能够直接解答或变换形式解答的练习题。

本书中，练习题和解题部分所占比重大。请各位阅读练习题时不要一带而过，要尽力去解答这些习题。只有这样才能与书中正确解法相比较，从而学到更多。

大家解题后最好把自己编写的代码跟书中代码进行比较。因为本书在确保效率的前提下，尽量精简了代码。各位可比较书中代码是否与自己编写的代码相似，实现算法的方法是否相同，进而可再思索一下是否有更好的实现方法。

必要的基础知识

本书面向计算机专业本科生。读者只要具备高中数学、计算机科学的入门级知识就足够了。所以，已经学过高中数学课程以及能够使用 C++^① 和 STL 就可以放心大胆地阅读本书。

给初学者的一点建议

本书涵盖多个主题，有些部分必然会先涉及尚未介绍的技巧或工具。^② 同样，本书不少示例也是利用后半部分要介绍的方法进行解答的。建议各位不要一味地按照章节顺序阅读，而是要先理解最重要、最基础的主题部分，等学到这些方法后再进行补充。建议初学者在深入学习之前，先阅读内容出现频率较高的几个章节。表 1-1 是推荐给初学者的课程表。

表 1-1 初学者的课程表

章节序号	章节题目
2	解决问题概述
3	编码与调试
4	分析算法的时间复杂度

① 本书中的源代码秉承 C++03 标准，而不是最新的 C++11 标准。

② 学习第 28 章后会理解，因为各个主题间的依赖性图表中存在循环。