

国际大学生 程序设计竞赛例题解 (五)

广东省大学生程序设计竞赛试题
(2006 - 2007年)

郭嵩山 张子臻 王磊 汤振东 著

含光盘1张



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国际大学生程序设计竞赛例题解

(五)

广东省大学生程序设计竞赛试题

(2006—2007 年)

郭嵩山 张子臻 王 磊 汤振东 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书收录了 2006—2007 年广东省大学生程序设计竞赛 (GDCPC 2006—2007, 即 ACM/ICPC 广东省赛) 的全部试题、完整的测试数据和答案, 以及 2004 年中山大学 ACM 集训队内部选拔赛题目、完整的测试数据和答案。为了方便读者学习, 本书对每个题目作了详尽的题目分析, 并详细地讲解其算法实现的原理, 同时提供了完善的参考程序及程序分析, 供读者参考。书中提供了竞赛时评判用的基本测试数据, 以方便读者测试自行完成上述题目的结果。随书附带的光盘中存放了竞赛时评判用的全部测试数据, 以便于有更高、更严格要求的同学能利用规模更大的测试数据进行训练和学习。

本书所提供的题目都是原创题, 题目构思新颖, 内容有趣。所涉及到的算法知识面广, 基本上覆盖大学计算机类本科专业所学到的基本算法。本书可以作为高等院校有关专业的研究生和本科学生参加国际大学生程序设计竞赛的辅导教材, 也可作为高等院校有关专业相关课程的教学参考书与例题集, 也特别适用于作为中学青少年信息学奥林匹克竞赛省级及省级以上优秀选手备战信息学奥林匹克竞赛的培训教材及训练题集。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

国际大学生程序设计竞赛例题解. 5, 广东省大学生程序设计竞赛试题: 2006—2007 年 / 郭嵩山等著. —北京: 电子工业出版社, 2008.11

ISBN 978-7-121-07435-6

I. 国… II. 郭… III. 程序设计—竞赛—高等学校—解题 IV. TP311.1-44

中国版本图书馆 CIP 数据核字 (2008) 第 148936 号

责任编辑: 龚立堇

印 刷: 北京市李史山胶印厂
装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 18 字数: 449.28 千字

印 次: 2008 年 11 月第 1 次印刷

印 数: 5 000 册 定价: 35.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

ACM 国际大学生程序设计竞赛 (ACM International Collegiate Programming Contest, 简称 ACM/ICPC) 是由国际计算机界历史悠久、颇具权威性的组织 ACM 学会 (Association for Computer Machinery) 主办的, 是世界上公认的规模最大、水平最高的国际大学生程序设计竞赛, 其目的旨在使大学生运用计算机来充分展示自己分析问题和解决问题的能力。该项竞赛从 1970 年举办至今已历 32 届, 因历届竞赛都荟萃了世界各大洲的精英, 云集了计算机界的“希望之星”, 而受到国际各知名大学的重视, 并受到全世界各著名 IT 企业的高度关注, 成为世界各国大学生最具影响力的国际级计算机类的赛事。ACM 所颁发的获奖证书也为世界各知名大学、各著名 IT 企业所认可。该项竞赛分为区域预赛和世界决赛两个阶段进行, 各预赛区第一名自动获得参加世界决赛的资格, 世界决赛安排在每年的 3~4 月举行, 而区域预赛安排在上一年度的 9~12 月在各大洲举行。ACM/ICPC 的区域预赛是规模很大、范围很广的赛事。以 2007 年为例, 全世界有 83 个国家和地区、1821 所大学、超过 6700 支参赛队在六大洲的 37 个赛站中争夺世界决赛的 100 个名额, 其激烈程度可想而知。

与其他编程竞赛相比, ACM/ICPC 题目难度更大, 更强调算法的高效性, 不仅要解决一个指定的命题, 而且必须以最佳的方式解决指定的命题。它涉及的知识面广, 与大学计算机系本科及研究生的课程直接关联, 如程序设计、离散数学、数据结构、人工智能、算法分析与设计等课程; 对数学要求更高; 由于采用英文命题, 对英语要求较高; ACM/ICPC 采用 3 人合作、共用一台电脑, 所以它更强调团队协作精神; 由于许多题目并无现成的算法, 需要具备创新的精神, ACM/ICPC 不仅强调学科的基础, 更强调全面素质和能力的培养; 由于 ACM/ICPC 是采用 5 小时全封闭式竞赛, 参赛队员与外界完全隔离, 独立完成, 是参赛队员实际能力的真实表露, 其成绩可信度甚高。ACM/ICPC 又是一种“开卷考试”, 可以带任何书籍、资料甚至源程序代码清单 (但不能带电子媒体), 不需要死背算法, 而强调的是算法的灵活运用; 与其他计算机竞赛 (如软件设计, 网站设计等) 相比, ACM/ICPC 有严谨而客观的评判规则 (严格的数据测试), 排除了因评委的主观因素而造成评审不公平的现象, 所以, ACM/ICPC 对成绩的争议较少。

中山大学自 1997 年首次参加 ACM/ICPC 亚洲区预赛以来的 11 年中, 每年都派出多支队伍共参加过 38 次亚洲区预赛, 成绩有 34 次在前 6 名之列, 其中有 21 次进入三甲, 夺得 4 次冠军 (1999 年台北, 2002、2003 年高雄, 2007 年岷港)、6 次亚军 (2000 年香港、筑波, 2003 年北京、广州, 2006 年河内, 2007 年首尔)、11 次季军 (1998—2000 年上海、2001 年达卡、2002 年北京, 2003 年高雄, 2004 年马尼拉, 2005 年台北、北京, 2006 年首尔, 2007 年成都); 中山大学的参赛队 9 次进入全球总决赛 (1999—2001 年、2003—2008 年): 2000 年在美国佛罗里达州奥兰多市举行的第 24 届全球总决赛中夺得了第 11 名的好成绩; 2001 年在加拿大温哥华市举行的第 25 届全球总决赛中首获铜牌 (世界第 14 名); 2003 年在美

国洛杉矶市好莱坞举行的第 27 届全球总决赛中获得世界第 8 名并首获银牌的好成绩,跻身世界八强之列;2004 年在捷克布拉格市举行的第 28 届全球总决赛中获得世界第 11 名并再获铜牌,且在中国内地高校中排名第一;2005 年在上海市举行的第 29 届全球总决赛中获得世界第 17 名;2006 年在美国得克萨斯州圣安东尼奥市举行的第 30 届全球总决赛中获得世界第 19 名;2007 年在日本东京市举行的第 31 届全球总决赛中获得世界第 26 名。2008 年在加拿大班夫市举行的第 32 届全球总决赛中获得世界第 23 名。

为了帮助高等院校的大学生们备战国际大学生程序设计竞赛,帮助他们提高程序设计水平和培养更强的分析问题与解决问题的能力,我们编写了这套《国际大学生程序设计竞赛例题解》。本书是这套《国际大学生程序设计竞赛例题解》的第五册,编程所用的语言版本是 Microsoft Visual C++6.0。本书全书共分 5 章,收录了 2006—2007 年广东省大学生程序设计竞赛(GDCPC 2006-2007,即 ACM/ICPC 广东省赛)的全部试题、完整的测试数据和答案,以及 2004 年中山大学 ACM 集训队内部个人选拔赛题、完整的测试数据和答案。为了方便读者学习,本书对每个题目作了详尽的题目分析,并详细地讲解其算法实现的原理,同时提供了完善的参考程序及其程序分析,供读者参考。书中提供了基本测试数据,以方便读者测试自行完成上述题目的结果。随书附带的光盘中存放所有例题中完整的测试数据,以便于有更高、更严格要求的同学能利用规模更大的测试数据进行训练和学习用。读者可利用《本书试题涉及知识点的说明》作为索引,根据知识点的分类进行学习,同时可根据目录中提供试题的难易程度,循序渐进地进行学习和训练。参与上述竞赛命题的有:莫瑜、黎俊瑜、蔡文志、关沛勇、张磊、林祺颖、梁锋、王颖、陈实、张子臻、王磊、张惠东、汤振东等,他们大部分是硕士研究生,都是参加过世界决赛或亚洲多个赛区区域预赛并取得很好成绩的中山大学队的主力队员。读者从附录中也可知道,中山大学 ACM/ICPC 队 11 年来取得了不俗的成绩,究其原因,除了有完善的选拔机制外,集训队内部个人选拔赛(俗称 4+2)的作用也是十分重要的。从本集开始,我们将逐步公开中山大学 ACM/ICPC 集训队内部选拔赛题目和题解,以便加强与读者的交流。

本书由中山大学郭嵩山教授和他指导的 3 位研究生撰写,郭嵩山老师是国际大学生程序设计竞赛中山大学队的主教练,张子臻、王磊、汤振东是参加过全球总决赛或亚洲多个赛区区域预赛并取得很好成绩的中山大学队的主力队员。我们期望能将自己的知识、经验、心得和体会,奉献给广大的程序设计爱好者,以便与大家共同探讨和交流。

本书所提供的题目全部为原创题,题目构思新颖、内容有趣,所涉及到的算法知识面广,基本上覆盖大学计算机类本科专业所学到的基本算法。本书可以作为高等院校大学生和研究生准备参加各级国际大学生程序设计竞赛活动的辅导教材及训练题集,也可以作为高等院校研究生和本科高年级学生学习相关课程的参考书,并适于作为中学省级及以上信息学奥林匹克优秀选手准备高层次程序设计竞赛的参考用书。

由于我们水平所限,书中难免有不足之处,欢迎读者批评指正,谢谢!

作者

2008 年 9 月

目 录

本书试题涉及知识点的说明	1
第 1 章 第 9 届中山大学程序设计竞赛预选赛 (2006 年) 试题分析	3
1.1 信息泛滥 (难度: ★★★☆☆)	3
1.1.1 试题	3
1.1.2 题目分析与算法实现	4
1.1.3 参考程序与程序分析	5
1.1.4 部分测试数据与输出结果	7
1.2 求和 (难度: ★★★★★☆)	7
1.2.1 试题	7
1.2.2 题目分析与算法实现	9
1.2.3 参考程序与程序分析	10
1.2.4 部分测试数据与输出结果	12
1.3 冲突 (难度: ★★★★★★)	13
1.3.1 试题	13
1.3.2 题目分析与算法实现	14
1.3.3 参考程序与程序分析	14
1.3.4 部分测试数据与输出结果	16
1.4 旅馆 (难度: ★★★☆☆☆)	17
1.4.1 试题	17
1.4.2 题目分析与算法实现	19
1.4.3 参考程序与程序分析	19
1.4.4 部分测试数据与输出结果	20
1.5 字符串子串 (难度: ★★☆☆☆☆)	21
1.5.1 试题	21
1.5.2 题目分析与算法实现	22
1.5.3 参考程序与程序分析	23
1.5.4 部分测试数据与输出结果	24
1.6 最大公约数 (难度: ★★★★★☆)	25
1.6.1 试题	25
1.6.2 题目分析与算法实现	25
1.6.3 参考程序与程序分析	26
1.6.4 部分测试数据与输出结果	28
1.7 木棒 (难度: ★☆☆☆☆☆)	29

1.7.1	试题	29
1.7.2	题目分析与算法实现	29
1.7.3	参考程序与程序分析	30
1.7.4	部分测试数据与输出结果	31
1.8	总体题目分析与比赛情况	31
1.8.1	题目总结分析	31
1.8.2	比赛情况	32
第2章	第4届广东省大学生程序设计竞赛暨第9届中山大学程序设计竞赛(2006年)	
	试题分析	33
2.1	过剩数(难度: ★☆☆☆☆)	33
2.1.1	试题	33
2.1.2	题目分析与算法实现	34
2.1.3	参考程序与程序分析	35
2.1.4	部分测试数据与输出结果	37
2.2	L形覆盖(难度: ★★☆☆☆)	38
2.2.1	试题	38
2.2.2	题目分析与算法实现	39
2.2.3	参考程序与程序分析	41
2.2.4	部分测试数据与输出结果	41
2.3	立方块(难度: ★★★★★☆)	42
2.3.1	试题	42
2.3.2	题目分析与算法实现	43
2.3.3	参考程序与程序分析	44
2.3.4	部分测试数据与输出结果	46
2.4	螺旋方阵(难度: ★★☆☆☆)	47
2.4.1	试题	47
2.4.2	题目分析与算法实现	48
2.4.3	参考程序与程序分析	49
2.4.4	部分测试数据与输出结果	50
2.5	电子书狂热者(难度: ★★★★★☆)	51
2.5.1	试题	51
2.5.2	题目分析与算法实现	54
2.5.3	参考程序与程序分析	55
2.5.4	部分测试数据与输出结果	57
2.6	栅格网络流(难度: ★★★★★★)	58
2.6.1	试题	58
2.6.2	题目分析与算法实现	60
2.6.3	参考程序及程序分析	61
2.6.4	部分测试数据与输出结果	65

2.7	连续边长多边形 (难度: ★★★☆☆)	66
2.7.1	试题	66
2.7.2	题目分析与算法实现	67
2.7.3	参考程序与程序分析	70
2.7.4	部分测试数据与输出结果	71
2.8	饥饿的母牛 (难度: ★★★☆☆)	72
2.8.1	试题	72
2.8.2	题目分析与算法实现	73
2.8.3	参考程序与程序分析	75
2.8.4	部分测试数据与输出结果	76
2.9	解析程序 (难度: ★★★☆☆)	76
2.9.1	试题	76
2.9.2	题目分析与算法实现	79
2.9.3	参考程序与程序分析	80
2.9.4	部分测试数据与输出结果	83
2.10	网络探测 (难度: ★★★☆☆)	84
2.10.1	试题	84
2.10.2	题目分析与算法实现	85
2.10.3	参考程序与程序分析	86
2.10.4	部分测试数据与输出结果	88
2.11	总体题目分析与比赛情况	88
2.11.1	题目总结分析	88
2.11.2	比赛情况	89
第3章	第10届中山大学程序设计竞赛预选赛 (2007年) 试题分析	90
3.1	圣诞节 (难度: ★★★☆☆)	90
3.1.1	试题	90
3.1.2	题目分析与算法实现	91
3.1.3	参考程序与程序分析	91
3.1.4	部分测试数据与输出结果	93
3.2	导弹 (难度: ★★☆☆☆)	94
3.2.1	试题	94
3.2.2	题目分析与算法实现	95
3.2.3	参考程序与程序分析	96
3.2.4	部分测试数据与输出结果	97
3.3	儿童节快乐 (难度: ★★★★★☆)	97
3.3.1	试题	97
3.3.2	题目分析与算法实现	98
3.3.3	参考程序与程序分析	99
3.3.4	部分测试数据与输出结果	102

3.4	排列 (难度: ★★★☆☆)	102
3.4.1	试题	102
3.4.2	题目分析与算法实现	103
3.4.3	参考程序与程序分析	104
3.4.4	部分测试数据与输出结果	104
3.5	士兵排队 (难度: ★★☆☆☆)	105
3.5.1	试题	105
3.5.2	题目分析与算法实现	106
3.5.3	参考程序与程序分析	106
3.5.4	部分测试数据与输出结果	107
3.6	我的总结 (难度: ★☆☆☆☆)	108
3.6.1	试题	108
3.6.2	题目分析与算法实现	109
3.6.3	参考程序与程序分析	109
3.6.4	部分测试数据与输出结果	110
3.7	准备工作 (难度: ★★★★★)	112
3.7.1	试题	112
3.7.2	题目分析与算法实现	113
3.7.3	参考程序与程序分析	114
3.7.4	部分测试数据与输出结果	116
3.8	总体题目分析与比赛情况	117
3.8.1	题目总结分析	117
3.8.2	比赛情况	118
第4章	第5届广东省大学生程序设计竞赛暨第10届中山大学程序设计竞赛(2007年)	
	试题分析	119
4.1	遥远的距离 (难度: ★★★★★☆)	119
4.1.1	试题	119
4.1.2	题目分析与算法实现	120
4.1.3	参考程序与程序分析	121
4.1.4	部分测试数据与输出结果	124
4.2	分数 (难度: ★☆☆☆☆)	125
4.2.1	试题	125
4.2.2	题目分析与算法实现	126
4.2.3	参考程序与程序分析	126
4.2.4	部分测试数据与输出结果	127
4.3	并行 (难度: ★★★★★☆☆)	127
4.3.1	试题	127
4.3.2	题目分析与算法实现	128
4.3.3	参考程序与程序分析	129

4.3.4	部分测试数据与输出结果	131
4.4	矩阵 (难度: ★★★★★☆)	131
4.4.1	试题	131
4.4.2	题目分析与算法实现	132
4.4.3	参考程序与程序分析	133
4.4.4	部分测试数据与输出结果	135
4.5	存在与否 (难度: ★★★★★★)	136
4.5.1	试题	136
4.5.2	题目分析与算法实现	138
4.5.3	参考程序与程序分析	139
4.5.4	部分测试数据与输出结果	142
4.6	树 (难度: ★★★★★☆)	144
4.6.1	试题	144
4.6.2	题目分析与算法实现	145
4.6.3	参考程序与程序分析	145
4.6.4	部分测试数据与输出结果	147
4.7	游戏 (难度: ★★☆☆☆☆)	147
4.7.1	试题	147
4.7.2	题目分析与算法实现	148
4.7.3	参考程序与程序分析	149
4.7.4	部分测试数据与输出结果	150
4.8	猴子和香蕉 (难度: ★★★★★☆)	151
4.8.1	试题	151
4.8.2	题目分析与算法实现	153
4.8.3	参考程序与程序分析	153
4.8.4	部分测试数据与输出结果	156
4.9	道路重建 (难度: ★★★★★☆)	157
4.9.1	试题	157
4.9.2	题目分析与算法实现	158
4.9.3	参考程序与程序分析	158
4.9.4	部分测试数据与输出结果	160
4.10	管道系统 (难度: ★★★★★☆)	161
4.10.1	试题	161
4.10.2	题目分析与算法实现	163
4.10.3	参考程序与程序分析	164
4.10.4	部分测试数据与输出结果	168
4.11	总体题目分析与比赛情况	169
4.11.1	题目总结分析	169
4.11.2	比赛情况	169

第5章	中山大学集训队内部选拔赛(2004年)试题分析	170
5.1	昵称(难度:★★★☆☆)	170
5.1.1	试题	170
5.1.2	题目分析与算法实现	171
5.1.3	参考程序与程序分析	171
5.1.4	部分测试数据与输出结果	173
5.2	三角形(难度:★★★☆☆)	174
5.2.1	试题	174
5.2.2	题目分析与算法实现	175
5.2.3	参考程序与程序分析	176
5.2.4	部分测试数据与输出结果	178
5.3	狙击兵(难度:★★★★☆)	179
5.3.1	试题	179
5.3.2	题目分析与算法实现	180
5.3.3	参考程序与程序分析	181
5.3.4	部分测试数据与输出结果	184
5.4	最小值(难度:★★★☆☆)	185
5.4.1	试题	185
5.4.2	题目分析与算法实现	186
5.4.3	参考程序与程序分析	187
5.4.4	部分测试数据与输出结果	189
5.5	最大的面积(难度:★★★★☆)	190
5.5.1	试题	190
5.5.2	题目分析与算法实现	191
5.5.3	参考程序与程序分析	191
5.5.4	部分测试数据与输出结果	196
5.6	最难的任务(难度:★☆☆☆☆)	197
5.6.1	试题	197
5.6.2	题目分析与算法实现	198
5.6.3	参考程序与程序分析	198
5.6.4	部分测试数据与输出结果	199
5.7	三只小猪(难度:★★★☆☆)	200
5.7.1	试题	200
5.7.2	题目分析与算法实现	201
5.7.3	参考程序与程序分析	202
5.7.4	部分测试数据与输出结果	205
5.8	基本的图问题(难度:★★★★☆)	206
5.8.1	试题	206
5.8.2	题目分析与算法实现	208

5.8.3	参考程序与程序分析	209
5.8.4	部分测试数据与输出结果	214
5.9	距离 (难度: ★★★☆☆)	215
5.9.1	试题	215
5.9.2	题目分析与算法实现	216
5.9.3	参考程序与程序分析	217
5.9.4	部分测试数据与输出结果	220
5.10	斐波纳契数 (难度: ★★☆☆☆)	221
5.10.1	试题	221
5.10.2	题目分析与算法实现	222
5.10.3	参考程序与程序分析	222
5.10.4	部分测试数据与输出结果	224
5.11	美味的蛋糕 (难度: ★★★☆☆)	224
5.11.1	试题	224
5.11.2	题目分析与算法实现	226
5.11.3	参考程序与程序分析	226
5.11.4	部分测试数据与输出结果	228
5.12	决斗俱乐部 (难度: ★★★☆☆)	229
5.12.1	试题	229
5.12.2	题目分析与算法实现	230
5.12.3	参考程序与程序分析	230
5.12.4	部分测试数据与输出结果	231
5.13	矩阵 (难度: ★★☆☆☆)	232
5.13.1	试题	232
5.13.2	题目分析与算法实现	233
5.13.3	参考程序与程序分析	234
5.13.4	部分测试数据与输出结果	235
5.14	石子游戏 (难度: ★★☆☆☆)	236
5.14.1	试题	236
5.14.2	题目分析与算法实现	237
5.14.3	参考程序与程序分析	238
5.14.4	部分测试数据与输出结果	238
5.15	构造序列 (难度: ★★☆☆☆)	239
5.15.1	试题	239
5.15.2	题目分析与算法实现	240
5.15.3	参考程序与程序分析	240
5.15.4	部分测试数据与输出结果	242
5.16	小树 (难度: ★★☆☆☆)	243
5.16.1	试题	243
5.16.2	题目分析与算法实现	244

5.16.3	参考程序与程序分析	244
5.16.4	部分测试数据与输出结果	246
5.17	数四边形 (难度: ★★★☆☆)	247
5.17.1	试题	247
5.17.2	题目分析与算法实现	248
5.17.3	参考程序与程序分析	248
5.17.4	部分测试数据与输出结果	249
5.18	最长数列 (难度: ★★★★★☆)	250
5.18.1	试题	250
5.18.2	题目分析与算法实现	251
5.18.3	参考程序与程序分析	251
5.18.4	部分测试数据与输出结果	254
5.19	原子能转变 (难度: ★★★☆☆)	254
5.19.1	试题	254
5.19.2	题目分析与算法实现	255
5.19.3	参考程序与程序分析	256
5.19.4	部分测试数据与输出结果	257
5.20	BOB 或 BBO (难度: ★★★★★☆)	258
5.20.1	试题	258
5.20.2	题目分析与算法实现	259
5.20.3	参考程序与程序分析	259
5.20.4	部分测试数据与输出结果	261
5.21	Turán 图 (难度: ★★★☆☆)	261
5.21.1	试题	261
5.21.2	题目分析与算法实现	262
5.21.3	参考程序与程序分析	263
5.21.4	部分测试数据与输出结果	263
5.22	金字塔 (难度: ★★★☆☆)	264
5.22.1	试题	264
5.22.2	题目分析与算法实现	265
5.22.3	参考程序与程序分析	266
5.22.4	部分测试数据与输出结果	268
附录 A	中国内地高校举办 ACM/ICPC 亚洲区预赛的情况	270
附录 B	中国内地高校参加 ACM/ICPC 全球总决赛的成绩	271
附录 C	中山大学队参加 ACM/ICPC 亚洲区域赛历年成绩表(1997—2007 年)	272
附录 D	中山大学队参加 ACM/ICPC 总决赛历年成绩表(1999—2008 年)	273
参考文献		274
作者简介		275

本书试题涉及知识点的说明

本书中的题型所覆盖的知识面相当广泛,包括了多种常用的算法设计方法,数据结构,图论,组合数学,离散数学和计算几何等方面的知识。

1. 算法设计方法

(1) 贪心方法与构造方法。具体策略是不从整体最优上加以考虑,而是选取某种意义上的局部最优解,当然很多时候贪心法所得到的结果也是整体最优的。实际上,贪心法得到的解往往也是通过构造解得出来的。例如本书中的“2.7 连续边长多边形”、“5.15 构造序列”、“5.21 Turán 图”和“5.22 金字塔”。

(2) 动态规划。动态规划是解决多阶段规划决策最优化问题的一种方法,其特点是将问题的解决过程划分为若干阶段,按照阶段进行状态转移,同时状态必须具有无后效性,即后面阶段的状态不会反过来影响前面阶段的状态。例如本书中的“1.4 旅馆”、“2.5 电子书狂热者”、“2.10 网络探测”、“3.2 导弹”、“4.5 存在与否”、“5.4 最小值”和“5.12 决斗俱乐部”。

(3) 搜索方法。搜索技术通常包括宽度优先搜索和深度优先搜索,在其他算法设计方法难以奏效的情况下,搜索通常是解决问题的有效手段。搜索是对问题的解空间进行遍历的过程,它从某一初始解出发,通过一些邻域扩展手段来不断产生新的解,从而达到遍历解空间的目的。当然有时对于问题解空间相当庞大的情况,完全遍历解空间是不现实的,此时就必须充分发掘问题所包含的约束条件,在搜索过程中应用这些条件来进行剪枝,从而减少搜索量。例如本书中的“2.3 立方块”、“3.7 准备工作”、“4.10 管道系统”和“5.11 美味的蛋糕”。

2. 数据结构

算法和数据结构是计算机程序中最基本的两个部分,如果说算法是对数据的处理,那么数据结构则是对数据的存储和组织。除了常用的简单数据结构如数组、链表、堆栈、队列等,还包括较为复杂的数据结构,如最小堆、AVL 树(平衡二叉树)、胜者树、线段树、并查集和散列表等,它们在某些应用场合可以极大地提高程序效率。例如本书中的“1.1 信息泛滥”涉及了平衡树或字典树,“2.6 栅格网络流”使用堆来寻找最短路,“3.3 儿童节快乐”涉及了线段树的使用,“4.4 矩阵”需要利用集合、映射的思想才能高效的设计出程序,“4.8 猴子和香蕉”涉及到最小堆操作,“5.1 昵称”可以使用哈希方法进行散列,“5.8 基本的图问题”涉及到线段树与并查集,“5.18 最长数列”需要对数据进行查找、插入、删除操作,可采用不同的数据结构实现,“5.20 BOB 或 BBO”蕴涵了字符串匹配的 KMP 思想。

3. 图论

图论研究的是顶点集与边集之间的关联关系，覆盖的知识点相当广泛，本书涉及的有：

- (1) 最短路问题，例如“4.9 道路重建”和“5.6 最难的任务”；
- (2) 网络流问题，例如“2.6 栅格网络流”和“5.3 狙击兵”；
- (3) 二部图匹配问题，例如“3.1 圣诞节”；
- (4) 树的遍历，例如“4.7 游戏”和“5.16 小树”；
- (5) 树的最近公共祖先问题，例如“5.9 距离”；
- (6) 最优哈密尔顿圈问题，例如“4.5 存在与否”。

4. 组合数学

本书试题主要涉及利用递推的思想求一些组合计数的问题，例如“3.4 排列”、“4.6 树”、“5.7 三只小猪”和“5.14 石子游戏”。

5. 离散数学

本书试题涉及了闭包运算，例如“1.3 冲突”。

6. 计算几何

本书试题涉及了计算几何中的一些基础的知识点，包括：

- (1) 复杂图形面积的计算，例如“2.8 饥饿的母牛”；
- (2) 求最远点对（旋转卡壳法），例如“4.1 遥远的距离”；
- (3) 求点集包含的最大面积，例如“5.5 最大面积”。

7. 数论

数论是一门研究整数性质的学科，本书试题涉及的知识点包括：

- (1) 质数的相关知识，例如“2.1 过剩数”；
- (2) 欧拉函数的应用，例如“1.6 最大公约数”；
- (3) 不定方程解的情况，例如“5.19 原子能转变”；
- (4) 矩阵快速幂取模，例如“5.10 斐波纳契数”。

8. 其他

此外本书试题还涉及其他一些不易归入上述类型的知识点：

- (1) 模拟，例如“2.9 解析程序”；
- (2) 排序，例如“1.5 字符串子串”和“4.3 并行”；
- (3) 统计，例如“1.2 求和”、“3.6 我的总结”、“4.2 分数”、“5.2 三角形”和“5.17 数四边形”；
- (4) 随机化算法，例如“5.13 矩阵”；
- (5) 趣味数学，例如“1.7 木棒”、“2.2 L 形覆盖”、“2.4 螺旋方阵”和“3.5 士兵排队”。

第 1 章

第 9 届中山大学程序设计竞赛预选赛

(2006 年) 试题分析

1.1 信息泛滥 (难度: ★★★★★)

1.1.1 试题

Problem A

Message Flood

(Input File: message.in / Output File: message.out)

Well, how do you feel about mobile phone? Your answer would probably be something like that “It’s so convenient and benefits people a lot”. However, if you ask Merlin this question on the New Year’s Eve, he will definitely answer “What a trouble! I have to keep my fingers moving on the phone the whole night, because I have so many greeting messages to send!”. Yes, Merlin has such a long name list of his friends, and he would like to send a greeting message to each of them. What’s worse, Merlin has another long name list of senders that have sent message to him, and he doesn’t want to send another message to bother them Merlin is so polite that he always replies each message he receives immediately). So, before he begins to send messages, he needs to figure to how many friends are left to be sent. Please write a program to help him.

Here is something that you should note. First, Merlin’s friend list is not ordered, and each name is alphabetic strings and case insensitive. These names are guaranteed to be not duplicated. Second, some senders may send more than one message to Merlin, therefore the sender list may be duplicated. Third, Merlin is known by so many people, that’s why some message senders are even not included in his friend list.

Input:

There are multiple test cases. In each case, at the first line there are two numbers n and m ($1 \leq n, m \leq 20000$), which is the number of friends and the number of messages he has received. And then there are n lines of alphabetic strings (the length of each will be less than 10), indicating the names of Merlin’s friends, one per line. After that there are m lines of alphabetic strings, which are the names of message senders.

The input is terminated by $n=0$.

Output:

For each case, print one integer in one line which indicates the number of left friends he must send.

Sample Input:

```
5 3
Inkfish
Henry
Carp
Max
Jericho
Carp
Max
Carp
0
```

Sample Output:

```
3
```

1.1.2 题目分析与算法实现

本题是比赛中较为简单的一题，主要考察编程者对字符串的处理和统计能力。题目的做法有很多，如果对 C++ 中的 STL 程序库比较熟悉，编写起来将较为简易。

题目的大意是，Merlin 拥有很多的朋友，除夕晚上很多朋友给 Merlin 发了短信，Merlin 也回复了这些信息。Merlin 有一个朋友列表，他想给每一位朋友都发送新年问候，但不包括之前已回复信息的朋友。问，Merlin 需要给多少个朋友发送新年问候。

分析题目可知，只需统计 Merlin 已回复信息的朋友在他朋友列表中的数量，用总数量减去该值即可。有如下几种具体做法：

(1) 将朋友列表按字符串大小排序，通过二分查找对 m 条已回复信息的朋友名字在排序表中查询并标记，最后顺序寻找未被标记的朋友即可。排序的时间复杂度为 $O(L \cdot n \log n)$ ，查询的时间复杂度为 $O(L \cdot m \log n)$ ，总时间复杂度为 $O(L(n+m) \log n)$ 。这里 L 为字符串的长度。

(2) 利用 C++ 里 STL 的 set 或 map 函数，这两个函数的底层都是用红黑树实现的，使用起来十分方便。将朋友列表名字顺次插入，然后将已回复信息的朋友在红黑树中删除，最后输出红黑树的结点数即可。其总时间复杂度也是 $O(L(n+m) \log n)$ 。

(3) 建立一棵字典树。由于名字最多只有 26 个字符，该字典树可以是一棵 26 叉树，每个字符串都可以由一条从根到某个叶子结点走出的路径所表示。

图 1.1.1 所示为一棵字典树，结点上的标号表示根到该结点经过的路径是否对应一个字符串，该图表示的字符串集合为 {“a”，“aap”，“ab”，“b”，“zb”}。

建立以朋友列表名字构成的字典树，再在字典树中抹去已回复信息的朋友的字符串，