

谨以此书献给上海交通大学获得ACM-ICPC世界冠军十周年

ACM国际大学生程序设计竞赛（ACM-ICPC）系列丛书

ACM国际大学生 程序设计竞赛 知识与入门

俞 勇 主编



清华大学出版社



ACM国际大学生程序设计竞赛（ACM-ICPC）系列丛书

ACM国际大学生 程序设计竞赛 知识与入门

俞 勇 主编



清华大学出版社
北 京

内 容 简 介

ACM 国际大学生程序设计竞赛 (ACM-ICPC) 是国际上公认的水平最高、规模最大、影响最深的计算机专业竞赛, 目前全球参与人数达 20 多万。本书作者将 16 年的教练经验与积累撰写成本系列丛书, 全面、深入而系统地将 ACM-ICPC 展现给读者。本系列丛书包括《ACM 国际大学生程序设计竞赛: 知识与入门》、《ACM 国际大学生程序设计竞赛: 算法与实现》、《ACM 国际大学生程序设计竞赛: 题目与解读》、《ACM 国际大学生程序设计竞赛: 比赛与思考》等 4 册, 其中《ACM 国际大学生程序设计竞赛: 知识与入门》介绍了 ACM-ICPC 的知识及其分类、进阶与角色、在线评测系统; 《ACM 国际大学生程序设计竞赛: 算法与实现》介绍了 ACM-ICPC 算法分类、实现及索引; 《ACM 国际大学生程序设计竞赛: 题目与解读》为各类算法配备经典例题及题库, 并提供解题思路; 《ACM 国际大学生程序设计竞赛: 比赛与思考》介绍了上海交通大学 ACM-ICPC 的训练及比赛, 包括训练札记、赛场风云、赛季纵横、冠军之路、峥嵘岁月。

本丛书适用于参加 ACM 国际大学生程序设计竞赛的本科生和研究生, 对参加青少年信息学奥林匹克竞赛的中学生也很有指导价值。同时, 作为程序设计、数据结构、算法等相关课程的拓展与提升, 本丛书也是难得的教学辅助读物。

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

ACM 国际大学生程序设计竞赛: 知识与入门/俞勇主编. —北京: 清华大学出版社, 2012.12
ISBN 978-7-302-29490-0

I. ①A… II. ①俞… III. ①程序设计-竞赛-高等学校-教学参考资料 IV. ①TP311.1

中国版本图书馆 CIP 数据核字 (2012) 第 163703 号

责任编辑: 龙启铭

封面设计: 傅瑞学

责任校对: 焦丽丽

责任印制: 沈 露

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 装 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185mm×230mm

印 张: 13.5

字 数: 280 千字

版 次: 2012 年 12 月第 1 版

印 次: 2012 年 12 月第 1 次印刷

印 数: 1~3000

定 价: 29.00 元

产品编号: 047606-01

写在最前面的话

自从上海交通大学 2002 年第一次、2005 年第二次获得 ACM 国际大学生程序设计竞赛（ACM International Collegiate Programming Contest，简称 ACM-ICPC 或 ICPC）世界冠军以来，总有记者邀请编者撰写冠军之路类的文章，也总有出版社希望编者出版 ACM-ICPC 竞赛类的书籍，因为没有想清楚怎么写，所以一直没动笔。直到 2010 年上海交通大学第三次获得 ACM-ICPC 世界冠军后，编者决定出版一套系列丛书，包括《ACM 国际大学生程序设计竞赛：知识与入门》、《ACM 国际大学生程序设计竞赛：算法与实现》、《ACM 国际大学生程序设计竞赛：题目与解读》及《ACM 国际大学生程序设计竞赛：比赛与思考》4 册书籍，全面、深入而系统地将 ACM-ICPC 展现给读者，把上海交通大学十多年来对 ACM-ICPC 竞赛的感悟分享给读者。

编写此系列丛书的另一个重要原因是 ACM-ICPC 竞赛在中国大陆的迅猛发展。自从 1996 年 ACM-ICPC 引入中国大陆，前六届仅设立 1 个赛区，目前每年一般设立 5 个赛区，并已有 30 所高校承办过亚洲区预赛；参赛学校从不满 20 所，到如今已达 200 多所；参赛人数从不到 100 人，到如今已超过 12 万人次；总决赛名额从起初的 3 个，到如今已超过 15 个。同时，中国大陆在 ACM-ICPC 竞赛上所取得的成绩也举世瞩目。清华大学 9 次获得总决赛奖牌（3 金 5 银 1 铜），位居奖牌榜之首，是实力最强、表现最稳定的高校；上海交通大学 8 次获得总决赛奖牌（4 金 3 银 1 铜），3 次夺得世界冠军，算是目前国内成绩最好的高校；中山大学 4 次获得总决赛奖牌（2 银 2 铜），在生源不占优势的情况下，这一成绩令人敬佩；复旦大学 3 次获得总决赛奖牌（1 银 2 铜），是公认的强校；浙江大学 2 次获得总决赛奖牌（1 金 1 银），1 次夺得世界冠军，再次让国人欢欣鼓舞；北京大学 1 次获得总决赛奖牌（1 铜），队员的综合实力堪称一流；最难能可贵的是，华南理工大学也获得过总决赛的奖牌（1 铜），它告诉我们，ACM-ICPC 不仅仅是“强校”之间的“对话”，只要坚持参与就会斩获成果。另外，至今已有 37 所大陆高校参加过全球总决赛，且不论成绩如何，他们在赛场上的奋斗亦值得称道。

本系列丛书的第一册《ACM 国际大学生程序设计竞赛：知识与入门》分为三个部分。知识点部分基本涵盖了竞赛中所涉及的主要知识点，包括数学基础、数据结构、图论、计算几何、论题选编、求解策略等六大类内容。入门与进阶部分介绍了包括如何快速入门、如何提高自身以及团队水平等，主要根据上海交通大学 ACM-ICPC 队多年参赛经验总结而来。在线资源部分对一些常用的在线评测系统和网上比赛进行了介绍。

本系列丛书的第二册《ACM 国际大学生程序设计竞赛：算法与实现》涵盖了大部分 ACM-ICPC 竞赛常用的经典算法，包括数学、图论、数据结构、计算几何、论题选编五个大类，对每个算法的代码实现，都配有接口说明以及简略的算法阐述，并提供算法的完整程序。并收集了一些实用的知识点及积分表，方便读者查找使用。

本系列丛书的第三册《ACM 国际大学生程序设计竞赛：题目与解读》分为两个部分。例题精讲部分针对第二册《ACM 国际大学生程序设计竞赛：算法与实现》中的算法配备经典例题，并提供细致的解题思路，读者可以通过这一部分学习和掌握算法；海量题库部分按照算法分类罗列出大量习题，并提供相应的题解，读者可以利用这一部分的题目进行训练，更加熟练地运用各类算法。

本系列丛书的第四册《ACM 国际大学生程序设计竞赛：比赛与思考》从 120 多名队员、2400 余篇文档中精心挑选、编纂而成的文集，包括训练札记、赛场风云、赛季纵横、冠军之路、峥嵘岁月，集中展现了上海交通大学 ACM-ICPC 队 16 年的奋斗历程，记载了这些队员为了实现自己的梦想而不懈努力、勇于拼搏的故事。

这是一套全面、系统地学习 ACM-ICPC 竞赛的知识类书籍；

这是一套详尽、深入地熟悉 ACM-ICPC 竞赛的算法及题目的手册类书籍；

这是一套程序设计、数据结构、算法等相关课程的拓展与提升类书籍；

这是一部上海交通大学 ACM-ICPC 队的成长史；

这是一部激励更多学子勇敢追寻并实现自己最初梦想的励志书。

历时 2 年零 5 个月，终于完成了本系列丛书，编者与队员有一种如释重负的感觉，因为我们把出版这套丛书看得很重，这是我们 16 年的经验与积累，希望对广大读者有用。

值此 ACM-ICPC 进入中国大陆 16 周年、上海交通大学获得 ACM-ICPC 世界冠军 10 周年之际，谨以此系列丛书——

纪念我们曾经走过的路、度过的岁月；

献给所有支持、帮助过我们的人……

俞 勇

2012 年 10 月于上海

前言

ACM 国际大学生程序设计竞赛 (ACM International Collegiate Programming Contest, 简称 ACM-ICPC 或 ICPC) 的试题覆盖了计算机科学以及相关数学领域众多知识点。这些知识通常会散落在各种书籍和论文之中, 学习和查找起来相对麻烦。此外, 竞赛中所考察的内容具有一定的特殊性, 即使是像由 Thomas H. Cormen 等编著的 *Introduction to Algorithms* 这样的经典书籍, 其中介绍的东西也并非都是竞赛中的考察点。因此能够有一本书针对 ACM-ICPC 竞赛所经常考察的知识点进行统一的介绍是有必要的。

本书分为三个部分, 第一部分为入门与进阶, 第二部分为知识点与求解策略, 第三部分为在线资源。第一部分介绍的内容包括如何快速入门、如何提高自身以及团队水平等, 主要是编者根据多年的参赛经验总结而来。第二部分基本涵盖了竞赛中所涉及的主要知识点, 包括数学基础、数据结构、图论、计算几何、论题选编、求解策略等 6 个大类。第三部分对一些常用的在线评测系统和网上比赛进行了介绍。其中, 第一部分和第三部分主要针对初学者。

由于 ACM-ICPC 竞赛涉及到的知识点较多, 很难在有限的篇幅里做到完全的覆盖, 编者主要根据重要性和实用性对内容进行了筛选。例如, 在介绍平衡二叉树时, 一般的数据结构书籍往往会介绍 AVL 树或是红黑树, 但本书只介绍了理解和实现起来更为简单的伸展树和 Treap。

本书知识点部分的内容采用的是一种类似百科全书的组织方式, 所以并不需要一章一章地从前往后阅读。读者完全可以选择一个自己感兴趣的知识点进行阅读, 并扩展延伸到与之相关的其他知识点中。

本书编写工作历时两年多, 参与编写工作的人员全部为上海交通大学 ACM-ICPC 队的现役与退役队员。他们参考了大量的书籍, 并结合了多年的竞赛经验, 对本书的内容进行选择、撰写和修改。

参与本书写稿、审稿的人员主要有 (按姓氏笔画为序): 乌辰洋、吴卓杰、张培超、陈彬毅、林承宇、易茜、郑翌、姜啸、曹正、曹雪智、商静波、彭上夫、程宇、谭天。

在此，衷心感谢所有为此书出版做出直接或间接贡献的人！也真心祝愿此书能够给更多读者带来学习知识的快乐！

由于时间仓促，作者水平有限，疏漏、不当和不足之处在所难免，真诚地希望专家和读者朋友们不吝赐教。如果您在阅读和使用此书过程中发现任何问题或有任何建议，恳请发邮件至 yyu@cs.sjtu.edu.cn，我们将不胜感激。

编 者

2012 年 10 月于上海

目 录

第一部分 入门与进阶

第 1 章 入门	3
1.1 ACM-ICPC 竞赛介绍	3
1.2 新手入门	5
1.3 团队的分工与配合	7
1.4 训练	9
1.5 备战分区赛	12
1.6 备战总决赛	13
第 2 章 进阶	16
2.1 如何提高读题能力	16
2.2 如何提高代码能力	17
2.3 Bug 与 Debug	19
2.4 从做题者到命题者	20
第 2 部分 知识点与求解策略	
第 3 章 数学基础	25
3.1 函数增长与复杂性分类	25
3.1.1 渐进符号	25
3.1.2 阶的计算	26
3.1.3 复杂性分类	27
3.2 概率论	28
3.2.1 事件与概率	28
3.2.2 期望与方差	30
3.3 代数学	31
3.3.1 矩阵	31
3.3.2 行列式	33
3.3.3 解线性方程组	34
3.3.4 多项式	37
3.3.5 复数	38
3.3.6 群	39
3.4 组合学	42
3.4.1 排列与组合	42
3.4.2 鸽巢原理	43
3.4.3 容斥原理	44
3.4.4 特殊计数序列	45
3.4.5 Pólya 计数定理	47
3.5 博弈论	50
3.5.1 博弈树	50
3.5.2 SG 函数	51
3.5.3 Nim 游戏与 Nim 和	53
3.6 数论	54
3.6.1 整除	54
3.6.2 不定方程	57
3.6.3 同余方程与欧拉定理	58
3.6.4 原根、离散对数和二项同余 方程	60
3.6.5 连分数	61
第 4 章 数据结构	64
4.1 线性表	64
4.1.1 链表	64
4.1.2 栈	65
4.1.3 队列	65
4.1.4 块状链表	66
4.2 集合	67
4.2.1 散列表	67
4.2.2 并查集	69
4.3 排序	71
4.3.1 朴素排序算法	71

4.3.1.1	插入排序.....	71	5.1.6	最短路.....	111
4.3.1.2	冒泡排序.....	72	5.1.6.1	Bellman-ford 算法.....	111
4.3.2	高效排序算法.....	73	5.1.6.2	Dijkstra 算法.....	113
4.3.2.1	归并排序算法.....	73	5.1.6.3	Floyd 算法.....	114
4.3.2.2	快速排序算法.....	74	5.2	树.....	115
4.3.2.3	线性排序算法.....	76	5.2.1	基本概念与遍历.....	115
4.4	树.....	78	5.2.1.1	树的基本定义与 术语.....	115
4.4.1	堆.....	78	5.2.1.2	树的遍历.....	117
4.4.1.1	二叉堆.....	78	5.2.2	生成树.....	117
4.4.1.2	左偏树.....	80	5.2.2.1	生成树的基本概念.....	117
4.4.2	二叉树.....	82	5.2.2.2	Prim 算法.....	118
4.4.2.1	二叉搜索树.....	82	5.2.2.3	Kruskal 算法.....	120
4.4.2.2	Treap.....	84	5.2.2.4	最小生成树的 变种.....	121
4.4.2.3	伸展树.....	85	5.2.2.5	生成树计数.....	123
4.4.3	线段树.....	89	5.3	二分图.....	124
第 5 章	图论.....	91	5.3.1	最大匹配.....	124
5.1	图.....	91	5.3.2	最大权匹配.....	126
5.1.1	基本概念.....	91	5.3.3	稳定婚姻.....	128
5.1.1.1	图的定义与基本 术语.....	91	5.4	网络流.....	129
5.1.1.2	匹配与覆盖.....	92	5.4.1	基本概念.....	129
5.1.1.3	独立集、团与支 配集.....	94	5.4.1.1	流网络.....	129
5.1.1.4	图的染色.....	95	5.4.1.2	残量网络.....	130
5.1.2	特殊图的分类.....	96	5.4.1.3	增广路径.....	130
5.1.3	图的遍历.....	99	5.4.1.4	最大流最小割 定理.....	131
5.1.3.1	深度优先遍历.....	99	5.4.2	最大流算法.....	131
5.1.3.2	广度优先遍历.....	100	5.4.2.1	Ford-Fulkerson 算法.....	131
5.1.4	连通性.....	103	5.4.2.2	Dinic 算法.....	133
5.1.4.1	连通性的基本 定义.....	103	5.4.3	费用流.....	135
5.1.4.2	割点与桥.....	104	5.4.4	流与割模型.....	137
5.1.4.3	强连通分量.....	105	5.4.4.1	上下界网络流.....	137
5.1.4.4	应用: 2-SAT.....	107	5.4.4.2	混合图欧拉回路.....	139
5.1.5	哈密顿路与欧拉路.....	108	5.4.4.3	最大权闭合子图.....	140
5.1.5.1	哈密顿路.....	108	第 6 章	计算几何.....	142
5.1.5.2	欧拉路.....	109	6.1	向量.....	142

6.2	点的有序化	143
6.3	多边形与圆	144
6.3.1	简单多边形	144
6.3.2	凸包问题	146
6.3.3	圆的面积并	147
6.4	半平面交	148
6.5	经典问题	151
6.5.1	线段求交	151
6.5.2	最近点对	152
6.5.3	最远点对	154
第 7 章	论题选编	156
7.1	背包问题	156
7.2	<i>LCA</i> 与 <i>RMQ</i>	157
7.3	快速傅里叶变换	159
7.4	字符串	161
7.4.1	字符串匹配	161
7.4.2	Trie	164
7.4.3	AC 自动机	165
7.4.4	后缀数组	167
7.4.5	扩展 KMP	169

第 8 章	求解策略	171
8.1	搜索	171
8.2	分治	175
8.3	贪心	176
8.4	动态规划	179
8.5	随机化	183

第三部分 在线资源

第 9 章	在线评测系统	187
9.1	基本使用方法	187
9.2	USACO 介绍	190
9.3	CII 介绍	191
9.4	PKU 介绍	192
9.5	SGU 介绍	193
9.6	SPOJ 介绍	195
第 10 章	网上比赛	197
10.1	GCJ 介绍	197
10.2	TopCoder 介绍	199
10.3	Codeforces 介绍	200
参考文献		203

第一部分

入门与进阶

1.1 ACM-ICPC 竞赛介绍

【概况】

ACM 国际大学生程序设计竞赛 (International Collegiate Programming Contest, ACM-ICPC 或 ICPC) 是由美国计算机协会 (Association for Computing Machinery, ACM) 主办的, 一项旨在展示大学生创新能力、团队精神和在压力下编写程序、分析和解决问题能力的年度竞赛。经过近 30 多年的发展, ACM-ICPC 已经发展成为最具影响力的大学生计算机竞赛。

ACM-ICPC 最早始于 1970 年, 当时美国德州 A&M 大学举办了首届竞赛, 主办方是 UPE 计算机科学荣誉协会 Alpha 分会。作为一种发现和培养计算机科学这一新兴领域顶尖学生的全新方式, 该竞赛很快得到了美国和加拿大多所大学的积极响应。1977 年, 在 ACM 计算机科学会议期间举办了首次总决赛, 并演变成为目前一年一届的国际性比赛。

ACM-ICPC 最初几届的参赛队伍主要来自美国和加拿大, 后来逐渐发展成为一项世界范围内的竞赛。自从 ICPC 得到了 IBM 等大型 IT 公司的赞助之后, 规模开始增长迅速。1997 年, 总共有来自 560 所大学的 840 支队伍参加了比赛, 而到了 2004 年, 这一数字迅速增加到 840 所大学的 4109 支队伍, 并正在以每年 10%~20% 的速度持续增长。

从 20 世纪 80 年代开始, ACM 将 ICPC 的总部设在位于美国德克萨斯州的贝勒大学。在大赛举办的早期, 冠军多为美国或加拿大的大学获得。进入 20 世纪 90 年代后期以来, 俄罗斯和其他一些东欧国家的大学连夺数次冠军。来自中国大陆的上海交通大学代表队在 2002 年美国夏威夷第 26 届、2005 年中国上海举行的第 29 届以及 2010 年中国哈尔滨的第 34 届全球总决赛上三夺世界冠军。这也是目前为止亚洲大学在该竞赛上取得的最好成绩。赛事的竞争格局已经由最初的北美大学一枝独秀演变成目前亚欧对抗的局面。

【比赛规则介绍】

比赛采用组队参赛形式, 由三名选手组队参加, 使用同一台计算机协作进行比赛。比赛时间为 5 个小时。比赛题目一般为 10 道左右的全英文试题。队伍解决了一道试题之后,

可将其提交给评委，由评委判断其是否正确。把试题的解答提交裁判称为运行，每一次运行会被判为正确或者错误，判决结果会及时通知参赛队伍。若提交的程序运行不正确，则该程序将被退回给参赛队，参赛队可以进行修改后再次提交。

程序运行的返回结果有以下几种：

- **Accepted**: 程序通过了测试数据的测试。
- **Wrong Answer**: 程序输出与期望输出不符。
- **Time Limit Exceeded**: 程序运行时间超过限制。
- **Memory Limit Exceeded**: 程序使用内存超过限制。
- **Runtime Error**: 程序运行时出错，可能原因包括数组越界、除零等。
- **Presentation Error**: 程序输出格式错误，比如输出了多余的空行或空格。
- **Output Limit Exceeded**: 程序输出过长，原因包括嵌套在死循环中的输出语句等。

竞赛结束后，参赛各队以解出问题的多少进行排名，具体规则如下：

- (1) 正确解决题目数较多的参赛者，排名靠前。
- (2) 对于正确解决题目数相同的参赛者，总罚时少的参赛者排名靠前。

罚时的计算规则如下：

一场比赛中的总罚时是所有正确解决的题目的罚时之和，没有正确解决的题目不计罚时。对于正确解决的某道题目，这一题的罚时=从比赛开始到正确解决这道题目经过的分钟数+20分钟×正确解决之前错误的提交数。

例如，比赛8点开始，你在8:30分正确解决某道题，之前有两次错误提交，那么这一题的罚时是 $30 + 20 \times 2 = 70$ 分钟。如果此外你还做出另外一题，该题目罚时是110分钟，那么比赛总罚时是两题之和，即 $70 + 110 = 180$ 分钟。

比赛中会根据上述规则，提供实时的比赛排名（standing），选手可通过排名系统获得比赛的实时信息，不同的赛区给出的排名系统也会有所不同，提供给选手的信息量是不一样的，例如，有的赛区不会给出具体题目的解答信息（只显示每个队过了几道题）。

一般来说，为了增加比赛的悬念，比赛的排名系统会在比赛进行4个小时后停止实时更新，此时选手无法通过排名系统或者最后一个小时的比赛情况。

另外，比赛的过程中，选手每解决出一道题目，将获得相应的一个气球，不同颜色的气球对应不同的题目。

比赛的过程中，选手遇到问题，可以向裁判提问，裁判将根据比赛规则选择是否回答选手的提问。在比赛中，由于数据错误等原因，会出现裁判改判的情况，即将所有选手之前提交的程序进行重新测试，并给出新的返回结果。

1.2 新手入门

首先，这里的新手是指有一定程序设计和数据结构基础的大学生。有一部分学生，在初中或者高中就参加过奥林匹克程序设计竞赛，在这些方面应该已经打下了较好的基础。其他同学也可以通过在大学中学习程序设计和数据结构这两门课程，或是通过看书自学等方式来了解这方面的基础知识。

在掌握程序设计和数据结构的基础上，以下4个部分是入门必备的：算法、英语读题、练习以及讨论交流。如果在这些方面都达到了一定水准，那么已经可以报名参加校内选拔，准备分区赛了。

下面将一一阐述这4个部分。

【算法】

ACM-ICPC 比赛每道题目都需要自己设计对应的算法，算法的重要性不言而喻。数据结构课程中可能会提到算法的概念，但不够系统。推荐初学者阅读《算法导论》这本书，讲解清晰详尽，中英文版都有，一些基本算法都会附带代码。另外刘汝佳与黄亮编写的《算法艺术与信息学竞赛》这本书同样非常适合，它的特色是例题和习题非常多，难度分布均匀。

这两本书内容都比较多，新手并不需要全部读完，把握主干即可，重点了解所有的经典算法和部分例题，根据个人需求来做习题。之后需要做专项训练时，可以再回头完成某个专题的所有练习。通过这两本书，要求达到以下目标：

- (1) 掌握常用算法；
- (2) 了解经典算法、知识点框架；
- (3) 可以独立分析出一些题目的算法。

【英语读题】

ACM-ICPC 的题目都是英文试题，所以拥有良好的英语阅读水平、能快速把握题目意思是一种优势。一般的题目并不是直接给出一个问题，而是有很丰富的故事背景，在有限的时间里，你需要从中获取关键内容，并且抽象出来，这样才能更方便地分析出算法。值得注意的是，有些关键信息可能会隐藏在故事背景等容易忽略的地方，所以即使英语阅读能力很好的人，也存在因为忽略了一些细节而导致理解错题意的可能。一道题目若是理解错题意，不仅做不出来，还可能浪费大量时间去实现一个错误的算法，在赛场上这种错误十分致命。总的来说，读题就是要做到快和准，要在这两点中找到平衡。

读题首先英语基础是关键，可以把题目就当成英语阅读来做。同时，读题也非常考验一个人的细心和耐心。要提高读题能力，需要保持良好的阅读习惯，并且要进行一定时间

的练习。当队伍组成以后，如果一个队伍的读题水平不高，可以组织专门的读题训练来加强。

【练习】

ACM-ICPC 竞赛支持的语言有 C/C++ 和 Java，熟练地使用这些语言中的一种是必要条件。并不需要对语言很精通了才开始做题，对于语言学习来说，练习和熟悉往往是相辅相成的。语言中一些偏向工程性的部分不需要特别了解，但是对于经常用到的部分，必须十分熟练，不能有半点不清楚的地方。

学会了语言和算法之后，实战是非常重要的环节。USACO（美国信息学奥林匹克竞赛）是个不错的选择，网上关于它的资源也很全面，遇到不会的可以轻松地找到解答，在遇到新知识，新算法的时候可以查阅一些专业书籍学习，USACO 每章也附带介绍了算法。

当 USACO 基本做完的时候编程能力应该已经不成问题了，接下来就是学习和巩固自己的算法能力。可以去各大在线题库上练习，比如 POJ、ZJU 等找对应的题目来做，巩固自己对算法的了解。在本书第三部分中将有更加详细的介绍，读者可以根据自己的需求去选择。

练习的时候，可以尽可能多的参加各类网络比赛。很多 OJ（Online Judge）会不定期的举办网络比赛，POJ、ZJU、SGU 等 OJ 的比赛形式和 ACM-ICPC 竞赛完全一样。TopCoder、Google Code Jam 等比赛则是侧重于个人实力，对于提高算法能力有很大帮助。

可能有人会问：做多少道题目才足够？我想答案是因人而异的，没有人可以给出完美的答案，但可以说多多益善。做题其实就是一个发现不足，加强训练，提高能力，然后再发现新的不足这样的循环，螺旋式的上升。没有人能够说自己已经达到能力的上限无法提高了，一定都还有提升的空间。

当然，做题不仅看数量，也要看效果。一味地为了数量而大量做题，不能很好地总结、发现问题，效果不可能很好；为了一道题目钻研几天，从而弄清楚了一类算法，这也是一种很好的方法。每个人要找到适合自己的练习的方法，提高练习的效率。

【讨论交流】

练习是比较枯燥的，可以找自己的队友或是一些参加 ACM-ICPC 的同学，经常交流自己获取的新算法，遇到的新难题，往往既能增加知识也能得到乐趣。除了队友和身边的同伴，网络也是很好的交流平台，比如各大 OJ 和 TopCoder 的论坛。

除了算法的交流外，也要学着去看别人的代码。同样的题目和算法，100 个人有 100 种实现方式，吸取别人代码的优点引为己用，可以更快地提高。同时，看别人代码对于查错也很有帮助。将来组队之后，给队友的代码查错也是必要的能力之一。

讨论交流同样也注重效率，尤其是在比赛时。两个人之间如何描述题意、描述算法，都是有讲究的。语言要有组织，表达清晰到位，简明扼要，不能产生歧义。这些在平时的讨论中就要有意识地培养。

1.3 团队的分工与配合

ACM-ICPC 竞赛要求三位选手组队参加比赛,在比赛的过程中,团队的分工与配合尤为重要。由于每支队伍只允许使用一台机器,所以一支队伍在比赛中需要分配好三个队员各自的角色和任务,从而将团队的效率发挥到最高。队员的角色一般可分为队长、解题核心以及辅助;比赛中需要完成的任务一般可分为读题、准备、编程和查错。

【角色分工】

队长

队长的主要工作是统筹规划,制定队伍的计划,并做出决策,包括比赛场上以及场下的训练。

赛场上,队长需要根据赛场的情况,合理分配时间、安排队员各自的任务以及做题顺序,同时在队员做题不顺、队伍落后的时候,给予队员鼓励。要做好这一点,队长首先要对自己两名队员的实力和特点有深入的了解,了解当面对不同问题时,不同队员的发挥水平,换句话说,队长必须知道每道题目在什么状况下给谁做效果最佳。其次,队长要对题目有较准确的定位,在了解题意之后能较快、较准确的判断该题的难度、考察方向,从而大致确定应当何时安排何人去解决该题。最后,队长要对场上其他队伍的解题情况保持关注,从其他队伍完成题目的情况中可以了解到题目的难易分布,可以帮助自己对题目进行更准确的定位,从而更好地制订自己队伍的做题规划。一般来说,队长本身不承担主要的解题任务,但需要有强大的解题实力,能够在必要时挺身而出,承担解题任务。

赛场下,队长是介于教练和队员之间的角色,队长需要将队里的问题反映给教练,也要将教练的建议转告给队员。在队伍士气低靡的时候鼓励队友,在队友紧张或兴奋的时候能够稳住队友的情绪,将队伍保持在最好的状态。

总的来说,队长不一定是编写代码能力很强的人,但需要对队员、题目甚至对手有充分的了解,在场上需要保持头脑清醒,对比赛要有宏观上的掌控力,并且能有一定的预见性,在综合考虑后能做出最适合于自己队伍的决策。

解题核心

解题核心的主要工作是承担大部分的解题任务,稳定地输出代码,因此解题核心需要具有强大的解题实力。

对解题核心来说,最重要的就是代码实力,具体来说分为代码速度以及代码正确率,需要在这两者之间找到平衡。其次解题核心的稳定性以及心理素质也非常的重要,解题核心发挥的不稳定对整个队伍来说是很严重的问题,解题核心需要做到即使是在手上有好几个代码没有通过的情况下,仍然能保持稳定的状态继续写新的题目。