

- 适合语言零基础的初学者
- 算法竞赛主要知识点的入门与拓宽
- 近200道竞赛真题分析
- 实用主义的C++和STL讲解
- 简洁、清晰、高效的示例代码

算法竞赛 入门经典 (第2版)

刘汝佳◎编著

清华大学出版社



内 容 简 介

本书是一本算法竞赛的入门与提高教材,把 C/C++ 语言、算法和解题有机地结合在一起,淡化理论,注重学习方法和实践技巧。全书内容分为 12 章,包括程序设计入门、循环结构程序设计、数组和字符串、函数和递归、C++ 与 STL 入门、数据结构基础、暴力求解法、高效算法设计、动态规划初步、数学概念与方法、图论模型与算法、高级专题等内容,覆盖了算法竞赛入门和提高所需的主要知识点,并含有大量例题和习题。书中的代码规范、简洁、易懂,不仅能帮助读者理解算法原理,还能教会读者很多实用的编程技巧;书中包含的各种开发、测试和调试技巧也是传统的语言、算法类书籍中难以见到的。

本书可作为全国青少年信息学奥林匹克联赛(NOIP)复赛教材、全国青少年信息学奥林匹克竞赛(NOI)和 ACM 国际大学生程序设计竞赛(ACM/ICPC)的训练资料,也可作为 IT 工程师与科研人员的参考用书。

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

算法竞赛入门经典/刘汝佳编著. —2 版. —北京:清华大学出版社,2014

(算法艺术与信息学竞赛)

ISBN 978-7-302-35628-8

I. ①算… II. ①刘… III. ①计算机算法-教材 IV. ①TP301.6

中国版本图书馆 CIP 数据核字(2014)第 046697 号

责任编辑:朱英彪

封面设计:刘超

版式设计:文森时代

责任校对:王云

责任印制:李红英

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

印 刷 者:清华大学印刷厂

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×260mm 印 张:30.5 字 数:794 千字

版 次:2009 年 11 月第 1 版 2014 年 6 月第 2 版 印 次:2014 年 6 月第 1 次印刷

印 数:1~5000

定 价:49.80 元

推荐序一

《算法竞赛入门经典（第2版）》要面世了。一方面高兴，一方面也想借题发挥，这是因为近年来我和我的团队致力于研究计算机教育的改革，对于应该如何提升学生的思维能力和行动能力有了新的认识。当然我会把握“不要离题太远”。

在我的书案上常年摆着一本蓝皮的书《算法艺术与信息学竞赛》，这是刘汝佳与黄亮合写的书，2003年12月我怀着喜悦的心情给这本书写了一页纸的序言。今天，时隔十年，我又拿起笔来为汝佳的新书作序，想到信息学奥林匹克的魅力，看到我们的学生能够承担起普及的责任和水平，此时此刻我的欣喜之情难以言表。“青出于蓝更胜于蓝”是我们当老师的最大愿望和期盼。汝佳之所以能写出这种内容和内涵丰富，文字也很难表达的思维艺术之美的好书，在于他对于信息学竞赛的热爱和他在青少年中普及计算机知识的强烈的责任感。汝佳为人低调诚实，做事认真负责，最可贵之处是那种“打破砂锅问到底”的求真务实精神，还有就是愿意和善于与人合作共事，能够真心听取别人的意见。

刘汝佳在中学参加信息学奥赛，进入清华大学后作为主力队员参加过ACM/ICPC世界大学生程序设计大赛，在本科和读研期间又长期担任国际信息学奥林匹克中国队的教练。很早以前他就说过：想写一本“从入门开始就能陪伴着读者的书”，意思是书的作用不仅是答疑、解惑，更是像朋友和知己，同读者一起探讨和研究问题。我当然赞成著书者的境界，在我给本科生上“程序设计基础”课时的感悟是：教学相长，发挥学生在学习中的主体作用，激发兴趣和调动积极性，创造条件，使其参与课程内容的研讨，并提出宝贵意见，是成就精品课的必由之路。汝佳说：“书的第12章就像是一个路标，告诉你每条路通往怎样的风景，但是具体还得靠读者自己走过去，在走之前也需要自己选择。”话说得很到位。闪光的东西蕴含在解决问题时的那些思维之美中，精妙的解题思路和策略有时令人拍案叫绝，一路学一路体味赏心悦目的风景，没有不爱学和学不会之理。

学会编程是一件相当重要的事，我在清华上课时对学生说“这是你们的看家本事”。一个国家，一个民族，要想不落伍，要想跻身于世界民族之林，关键在于拥有高素质的人才。学习和掌握信息科学与技术，在高水准人才的知识结构中占有重要的地位。讲文化要以科学为基础，讲科学要提高到文化的高度。学习计算机必须了解这一学科的内在规律和特征，“构造性”和“能行性”是计算机学科的两个最根本特征。与构造性相应的构造思维，又称计算思维，指的是通过算法的“构造”和实现来解决一个给定问题的一种“能行”的思维方式。

有些问题没有固定的解法，给读者留有广阔的发挥创造力的空间，经过思考构造出的算法能不能高效地解决问题，都得通过上机实践的检验，在这一过程中思维能力和行动能力会同步提升。我认为高手应该是这样炼成的。光说不练，纸上谈兵是绝对学不会的。

当前，有识之士已经认识到：大学计算机作为基础课，与数学、物理同样重要。培养计算机的应用能力，掌握使用计算机的思路和方法，必须既动手又动脑，体会和感悟蕴含

于其中的计算思维要素，对于现代人是非常重要的。

当你拿到这本书时，建议你先看“阅读说明”。其中有两点，一是“本书最好是有人带着学习”，如果这一点做不到的话，建议你求助于网络，开展合作学习，向高人请教，让心得共享，这些都符合现代学习理念；二是“一定要重视书中的提示”，因为其中包含着需要掌握的重要知识点和编程技巧，你会发觉有些内容在一般教科书中是看不到的。

这是一本学习竞赛入门的书。我想说的是参加信息学竞赛入门不难，深造也是做得到的，关键是专心、恒心与信心，世上无难事，只要肯攀登！

全国信息学奥林匹克（NOI）科学委员会名誉主席
国际信息学奥林匹克中国队总教练
清华大学计算机科学与技术系
吴文虎

推荐序二

认识刘汝佳已有十多年的时间。2000年3月，我作为NOI（全国信息学奥林匹克）科学委员会委员赴澳门参加NOI2000的竞赛组织工作，正是在那届NOI上，刘汝佳以总分第四名的优异成绩获得NOI2000金牌并进入国家集训队。保送进入清华计算机系后，他又经选拔成为清华大学ACM队的主力队员，先后获得2001年ACM/ICPC（国际大学生程序设计竞赛）亚洲-上海赛区冠军和2002年世界总决赛的银牌（世界第四）。其后的多年时间里，他还同时担任NOI科学委员会的学生委员和IOI（国际信息学奥林匹克）中国国家队的教练。

2005—2007年，刘汝佳在清华计算机系读研期间，正逢我为本科生开设《数据结构》课程，他多次担任这门课程的助教工作。十多年来他几乎每年都受聘参加NOI冬令营的授课，并赢得听课选手的一致好评。

可以说，刘汝佳既是一名NOI和ACM/ICPC成绩优异的金牌选手，又是曾多年执教国家集训队的金牌教练，同时还作为在NOI冬令营等竞赛培训第一线参与授课培训最受欢迎的金牌教师。他在信息学奥赛方面的丰富经历与多重身份，特别是近20年来他对信息学奥赛的痴迷与执着，使得他对程序设计语言得心应手，对各种数据结构和算法的理解也颇有心得。这些都为他日后编写算法和编程竞赛的多部专著奠定了坚实的基础。

以信息学奥赛的应用为背景，将数据结构和算法的知识点讲解与信息学奥赛的问题求解紧密联系在一起，通过大量鲜活的奥赛解题实例让读者领悟到不同算法和数据结构的精妙，是刘汝佳教材的独到之处与鲜明特色。这也是国内大量单一身份的作者（课程教师）编写的教材所欠缺又无法企及的。我们通常看到的或者是单纯叙述算法与数据结构知识的普通教材，或者是专门针对竞赛题目的题解汇编，但真正既能涵盖算法竞赛的主要知识点，又融入大量比赛技巧和解题经验教训，且将二者融会贯通的教材实在是凤毛麟角。刘汝佳的教材在这方面或者可以说是填补了空白，至少也称得上是独树一帜，这也是许多作者心有余而力不足的。

从编排结构和写作特点上来说，刘汝佳的专著充分考虑不同层次的读者阅读需求，在《算法竞赛入门经典》中分为语言篇、基础篇和竞赛篇，循序渐进，既适合初学者，也适合高手进一步提升研读。特别是书中大量实用的示例代码和丰富的例题与习题，使各种水平的选手都能从他的书中获益并汲取营养，是有志在竞赛中取得佳绩的选手所必备的参考资料。

从刘汝佳的第一本专著《算法艺术与信息学竞赛》问世至今刚好十年时间。其间，在读者好评如潮和高于市场预期的销量面前，刘汝佳并没有就此止步，而是搜集信息学竞赛选手的各种需求，对已出版的教材进行更多有益的尝试。在繁忙的工作之余仍能挤出时间笔耕不辍，这也从一个侧面反映了他的勤奋刻苦、不懈进取以及对信息学奥赛的执着追求。我们为国内信息学奥赛领域有汝佳这样优秀的专家学者感到庆幸。中国计算机学会2009年



为他颁发的“特别贡献奖”实至名归。

今年正值邓小平同志提出“计算机的普及要从娃娃抓起”和全国信息学奥林匹克创办30周年，刘汝佳的新版专著为这一历史时刻增光添彩。我们期待信息学奥赛领域有更多更好的教材专著问世。让一切与信息学奥赛相关的劳动、知识、技术、管理和资本的活力竞相迸发，让一切与信息学奥赛创新改革的源泉充分涌流。让我们共同努力，谱写全国信息学奥林匹克的新篇章。

全国信息学奥林匹克（NOI）科学委员会主席
清华大学计算机科学与技术系
王 宏

推荐序三

ACM 国际大学生程序设计竞赛（简称为 ACM-ICPC 或 ICPC）始于 1970 年，成形于 1977 年，并于 1996 年进入我国大陆。由于该项赛事形式别具一格，竞赛题目既有挑战性又有趣味性，有助于培养参赛选手的抽象思维、逻辑思维、心理素质、团队合作和协同能力，所以深受参赛选手们的喜爱，ACM-ICPC 赛事也从不为人所知、从组委会千方百计邀请各个兄弟院校组队参赛捧场，到如今各赛区组委会都遇到了多次扩容仍无法满足大家的参赛愿望。虽然在 1996 年仅有 19 所学校的 25 队参赛，但在 2013 年已有来自 250 所高校的 4300 多队参加了网络赛，170 多所学校的 840 多队获得了参加现场赛的机会。从竞赛中脱颖而出的优秀选手也获得了国内外著名企业的高度认可，可以说 ACM-ICPC 大赛得到社会各界的广泛认可，这也是对我们这批该项赛事组织者而言最大的鼓励和回报。

刘汝佳则是从该项赛事中涌现出的佼佼者之一，他不仅在该项赛事中取得了优异的成绩，获得了 2011 年 ACM-ICPC 亚洲区上海赛区冠军和 2002 年夏威夷全球总决赛银奖第一，而且热心于该项赛事的著书写作和命题等工作。

《算法竞赛入门经典（第 2 版）》将程序设计语言和算法灵活地结合在一起，形式独特，算法部分讲解细致，内容涵盖了许多经典算法，强调了不少入门时的注意事项，并且在一定程度上回答了“参加 ACM-ICPC 需要掌握哪些基本的知识点、哪些经典算法、要注意哪些基本的编程技巧、ICPC 优秀选手是如何分析问题和优化代码的”等一系列问题。

虽然本书不是一本专门为 ACM-ICPC 而写的教材，但是书中所有例题都来自 ACM-ICPC 相关竞赛，不仅可作为 ACM-ICPC 的入门参考书，同时也是一本适合具有一定数学基础但没有接触过程序设计的大学生阅读的算法参考书。

ACM 国际大学生程序设计竞赛中国区指导委员会秘书长
上海大学
周维民

第 2 版前言

《算法竞赛入门经典》第 1 版出版至今已有四个年头。这四年间发生了很多变化，如 NOI 系列比赛终于对 STL “解禁”，如 C11 和 C++11 标准出台，g++ 编译器升级（直接导致本书第 1 版中官方使用的 `<?>` 和 `>?` 运算符无法编译通过），如《算法竞赛入门经典——训练指南》的出版弥补了本书第 1 版的很多缺憾，再如 ACM/ICPC 的蓬勃发展，使更多的大学生了解并参与到了算法竞赛中来……

看来，是时候给本书“升级”了。

主要的变化

我原本打算只是增加一章专门介绍 C++ 和 STL，用符合新语言规范的方式重写部分代码，顺便增加一些例题和习题，没想到一写就是 100 页——几乎让书的篇幅翻了一倍。写作第 1 版时，220 页的篇幅是和诸位一线中学教师商量后定下来的，因为书太厚会让初学者望而生畏。不过这几年的读者反馈让我意识到：由于篇幅限制，太多的东西让读者意犹未尽，还不如多写点。虽然之后出版了《算法竞赛入门经典——训练指南》，但那本书的主要目标是补充知识点，即拓展知识宽度，而我更希望在知识宽度几乎不变的情况下增加深度——我眼中的竞赛应该主要比思维和实践能力，而不是主要比见识。

索性，我继续加大篇幅，用大量的例子（包括题目和代码）来表现我想向读者传达的信息。一位试读的朋友在收到第一份书稿片段时惊呼：“题目的质量比第 1 版提高太多了！”这正是我这次改版的主要目的。

具体来说，这次改版有以下变化：

- 在前 4 章中逐步介绍一些更实用的语言技巧，直接使用竞赛题目作为例子。
- 全新的第 5 章，讲解竞赛中最常用的 C++ 语法，包括 STL 算法和容器。
- 第 6~7 章作为基础篇，加大代码和技巧的比例，并适当增加例题。
- 第 8~11 章作为中级篇，增加了各种例题，着重锻炼思维能力。
- 全新的第 12 章作为高级篇，在《算法竞赛入门经典——训练指南》的基础上补充少量知识点与大量精彩例题。

需要特别说明的是第 12 章出现的原因。这一章的内容很难，而且要求读者熟练掌握《算法竞赛入门经典——训练指南》的主要内容，看起来和“入门”二字是矛盾的。其实本书虽然名为“入门经典”，实际上却不仅只适合入门读者。根据这几年读者反馈的情况来看，有相当数量的有经验的选手也购买了本书。原因在于：很多有经验的选手属于“自学成才”，总觉得自己可能会漏掉点什么基础知识。事实也是如此：本书中提到的很多代码和分析技巧是传统教科书中见不到的，对于很多有经验的选手来说也是“新鲜事物”，并且他们能比初学者更快、更好地把这些知识运用到比赛中去。本书第 12 章就是为这些读者准备的。

如果这样解释还不够直观，就把第 12 章作为一个游戏里通关后多出来的 Hard 模式吧！

阅读说明

既然内容有了较大变化，阅读方式也需要再次说明一下。首先，和本书第 1 版一样，本书最好是有人带着学习，如老师、教练或者学长。随着网络的发展，这个条件越来越容易满足了——就算是没人指导，也可以在别人的博客中留言，或者在贴吧中寻求帮助。

一定要重视书中的“提示”。书中有很多“提示”部分都是非常重要的知识点或者技巧，有些提示看似平凡无奇，但如果没有引起重视而导致赛场上丢分，可是会追悔莫及的。

接下来是关于新增第 5 章的。首先声明一点，这一章并不是 C++ 语言速成——C++ 语言是不可能速成的。这一章不是说你从头读到尾然后就掌握 C++ 了，而是提供一个纲要，告诉你哪些东西是算法竞赛中最常用的，以及这些东西应当如何使用。你可以先另外找一本书（或者阅读网上的文章）学习 C++，然后再看本书第 5 章（目的是把那些又容易晕又不那么有用的知识从脑子里删除），也可以直接看本书第 5 章，每次遇到看不懂或者觉得不够详细的地方，再找其他参考书来学。顺便说一句，就算你已经非常熟悉 C++ 了，也最好浏览一下第 5 章（特别是代码！）。这不会花费太多时间，但很可能学到有用的东西。

忍不住再说点题外话。有时学习算法的最好方法并不是编写程序，而是手算。“手算”这个词听上去有点枯燥，改成“玩游戏”如何？如《雷顿教授与不可思议的小镇》就是一个不错的选择——它包含了过河问题（谜题 7、93）、找砝码（谜题 6、131）、一笔画（谜题 30、39）、 n 皇后（谜题 80~83、130）、倒水问题（谜题 23、24、78）、幻方（谜题 95）、华容道（谜题 97、132、135）等诸多经典问题。

致谢

虽然多出来了 200 多页，其实本书的改版工作并没有花费太长时间（不到半年），在此期间也没有麻烦太多朋友读稿和讨论。参与本书第 2 版读稿和校对工作的几位朋友分别是：陈锋（第 8~11 章）、王玉斌（第 8~9 章，第 12 章）、郭云韬（第 12 章）、曹海宇（第 5 章、第 9 章）、陈立杰（第 12 章）、叶子卿（第 12 章）、周以凡（第 12 章）。

感谢给我发邮件以及在 googlecode 的 wiki 中留言指出本书第 1 版勘误的网友们：imxivid、zr95.vip、李智维、王玉、chnln0526、yszhou4tech、metowolf88、zhongying822、chong97993、tplee923、wtx20074587、chu.pang 等，你们的支持和鼓励是我写作的重要动力。

另外，书中部分难题的题解离不开以下朋友的赐教和讨论：Md.Mahbubul Hasan、Shahriar Manzoor、Derek Kisman、Per Austrin、Luis Garcia、顾昱洲、陈立杰、张培超等。

第 2 版的习题全部（这次不仅仅是“主要”了）来自 UVa 在线评测系统，感谢 Miguel Revilla 教授、他的儿子 Miguel Jr. 和 Carlos M. Casas Cuadrado 对本书的大力支持。

最后，再次感谢清华大学出版社的朱英彪编辑在这个恰当的时机提出改版事宜，并容忍我把交稿时间一拖再拖。希望这次改版不会让你失望。

刘汝佳

前 言

“听说你最近在写一本关于算法竞赛入门的书？”朋友问我。

“是的。”我微笑道。

“这是怎样的一本书呢？”朋友很好奇。

“C 语言、算法和题解。”我回答。

“什么？几样东西混着吗？”朋友很吃惊。

“对。”我笑了，“这是我思考许久后做出的决定。”

大学之前的我

12 年前，当我翻开 Sam A. Abolrous 所著的《C 语言三日通》的第一页时，我不会想到自己会有机会编写一本讲解 C 语言的书籍。当时，我真的只用了 3 天就学完了这本书，并且自信满满：“我学会 C 语言啦！我要用它写出各种有趣、有用的程序！”但渐渐地，我认识到了：虽然浅显易懂，但书中的内容只是 C 语言入门，离实际应用还有较大差距，就好比小学生学会造句以后还要下很大工夫才能写出像样的作文一样。

第二本对我影响很大的书是 Sun 公司 Peter van der Linden (PvdL) 所著的《C 程序设计奥秘》。作者称该书应该是每一位程序员“在 C 语言方面的第二本书”，因为“书中绝大部分内容、技巧和技术在其他任何书中都找不到”。原先我只把自己当成是程序员，但在阅读的过程中，我开始渐渐了解到硬件设计者、编译程序开发者、操作系统编写者和标准制定者是怎么想的。继续阅读增强了我的领悟：**要学好 C 语言，绝非熟悉语法和语义这么简单。**

后来，我自学了数据结构，懂得了编程处理数据的基本原则和方法，然后又学习了 8086 汇编语言，甚至曾没日没夜地用 SoftICE 调试《仙剑奇侠传》，并把学到的技巧运用到自己开发的游戏引擎中。再后来，我通过《电脑爱好者》杂志上一则不起眼的广告了解到全国信息学奥林匹克联赛（当时称为分区联赛，NOIP 是后来的称谓）。“学了这么久的编程，要不参加个比赛试试？”想到这里，我拉着学校里另外一个自学编程的同学，找老师带我们参加了 1997 年的联赛——在这之前，学校并不知道有这个比赛。凭借自己的数学功底和对计算机的认识，我在初赛（笔试）中获得全市第二的成绩，进入了复赛（上机）。可我的上机编程比赛的结果是“惨烈”的：第一题有一个测试点超过了整数的表示范围；第二题看漏了一个条件，一分都没得；第三题使用了穷举法，全部超时。考完之后我原以为能得满分的，结果却只得了 100 分中的 20 多分，名落孙山。

痛定思痛，我开始反思这个比赛。一个偶然的机，我拿到了一本联赛培训教材。书上说，比赛的核心是算法（Algorithm），并且推荐使用 Pascal 语言，因为它适合描述算法。我复制了一份 Turbo Pascal 7.0（那时网络并不发达）并开始研究。由于先学的是 C 语言，

所以我刚开始学习 Pascal 时感到很习惯：赋值不是“=”而是“:=”，简洁的花括号变成了累赘的 begin 和 end，if 之后要加个 then，而且和 else 之间不允许写分号……但很快我就发现，这些都不是本质问题。在编写竞赛题的程序时，我并不会用到太多的高级语法。Pascal 的语法虽然稍微啰嗦一点，但总体来说是很清晰的。就这样，我只花了不到一天的时间就把语法习惯从 C 转到了 Pascal，剩下的知识就是在不断编程中慢慢地学习和熟练——学习 C 语言的过程是痛苦的，但收益也是巨大的，“轻松转到 Pascal”只是其中一个小小的例子。

我学习计算机，从一开始就不是为了参加竞赛，因此，在编写算法程序之余，我几乎总是使用熟悉的 C 语言，有时还会用点汇编，并没有觉得有何不妥。随着编写应用程序的经验逐渐丰富，我开始庆幸自己先学的是 C 语言——在我购买的各类技术书籍中，几乎全部使用的是 C 语言而不是 Pascal 语言，尽管偶尔有用 Delphi 的文章，但这种语言似乎除了构建漂亮的界面比较方便之外，并没有太多的“技术含量”。我始终保持着对 C 语言的熟悉，而事实证明这对我的职业生涯发挥了巨大的作用。

中学竞赛和教学

在大学里参加完 ACM/ICPC 世界总决赛之后（当时 ACM/ICPC 还可以用 Pascal，现在已经不能用了），我再也没有用 Pascal 语言做过一件“正经事”（只是偶尔用它给一些只懂 Pascal 的孩子讲课）。后来我才知道，国际信息学奥林匹克系列竞赛是为数不多的几个允许使用 Pascal 语言的比赛之一。IT 公司举办的商业比赛往往只允许用 C/C++ 或 Java、C#、Python 等该公司使用较为频繁的语言（顺便说一句，C 语言学好以后，读者便有了坚实的基础去学习上述其他语言），而在做一些以算法为核心的项目时，一般来说也不能用 Pascal 语言——你的算法程序必须能和已有的系统集成，而这个“现有系统”很少是用 Pascal 写成的。为什么还有那么多中学生非要用这个“以后几乎再也用不着”的语言呢？

于是，我开始在中学竞赛中推广 C 语言。这并不是说我希望废除 Pascal 语言（事实上，我希望保留它），而是希望学生多一个选择，毕竟并不是每个参加信息学竞赛的学生都将走入 IT 界。但如果简单地因为“C 语言难学难用，竞赛中还容易碰到诸多问题”就放弃学习 C 语言，我想是很遗憾的。

然而，推广的道路是曲折的。作为五大学科竞赛（数学、物理、化学、生物、信息学）中唯一一门高考中没有的“特殊竞赛”，学生、教师、家长所走的道路要比其他竞赛要艰辛得多。

第一，数理化竞赛中所学的知识，多是大学本科时期要学习的，只不过是提前灌输给高中生而已，但信息学竞赛中涉及的很多知识甚至连本科学生都不会学到，即使学到了，也只是“简单了解即可”，和“满足竞赛的要求”有着天壤之别，这极大地削减了中学生学习算法和编程的积极性。

第二，学科发展速度快。辅导信息学竞赛的教师常常有这样的感觉：必须不停地学习学习再学习，否则很容易跟不上“潮流”。事实上，学术上的研究成果常常在短短几年之内就体现在竞赛中。

第三，质量要求高。想法再伟大，如果无法在比赛时间之内把它变成实际可运行的程

序，那么所有的心血都将白费。数学竞赛中有可能在比赛结束前 15 分钟找到突破口并在交卷前一瞬间把解法写完——就算有漏洞，还有部分分数呢；但在信息学竞赛中，想到正确解法却 5 个小时都写不完程序的现象并不罕见。连程序都写不完当然就是 0 分，即使程序写完了，如果存在关键漏洞，往往还是 0 分。这不难理解——如果用这个程序控制人造卫星发射，难道当卫星爆炸之后你还可以向人炫耀说：“除了有一个加号被我粗心写成减号从而引起爆炸之外，这个卫星的发射程序几乎是完美的。”

在这样的情况下，让学生和教师放弃自己熟悉的 Pascal 语言，转向既难学又容易出错的 C 语言确实是难为他们了，尤其是在 C 语言资料如此缺乏的情况下。等一下！C 语言资料缺乏？难道市面上不是遍地都是 C 语言教材吗？对，C 语言教材很多，但和算法竞赛相结合的书却几乎没有。不要以为语言入门以后就能轻易地写出算法程序（这甚至是很多 IT 工程师的误区），多数初学者都需要详细的代码才能透彻地理解算法，只了解算法原理和步骤是远远不够的。

大家都知道，编程需要大量的练习，只看和听是不够的。反过来，如果只是盲目练习，不看不听也是不明智的。本书的目标很明确——提供算法竞赛入门所必需的一切“看”的蓝本。有效的“听”要靠教师的辛勤劳动，而有效的“练”则要靠学生自己。当然，就算是最简单的“看”，也是大有学问的。不同的读者，往往能看到不同的深度。请把本书理解为“蓝本”。没有一本教材能不加修改就适用于各种年龄层次、不同学习习惯和悟性的学生，本书也不例外。我喜欢以人为本，因材施教，不推荐按照本书的内容和顺序填鸭式地教给学生。

内容安排

前面花了大量篇幅讨论了语言，但语言毕竟只是算法竞赛的工具——尽管这个工具非常重要，却不是核心。正如前面所讲，算法竞赛的核心是算法。我曾考虑过把 C 语言和算法分开讲解，一本书讲语言，另一本书讲基础算法。但后来我发现，其实二者难以分开。

首先，语言部分的内容选择很难。如果把 C 语言的方方面面全部讲到，篇幅肯定不短，而且和市面上已有的 C 语言教材基本上不存在区别；如果只是提纲挈领地讲解核心语法，并只举一些最为初级的例子，看完后读者将会处于我当初 3 天看完《C 语言三日通》后的状态——以为自己都懂了，慢慢才发现自己学的都是“玩具”，真正关键、实用的东西全都不懂。

其次，算法的实现常常要求程序员对语言熟练掌握，而算法书往往对程序实现避而不谈。即使少数书籍给出了详细代码，但代码往往十分冗长，不适合用在算法竞赛中。更重要的是，这些书籍对算法实现中的小技巧 and 常见错误少有涉及，所有的经验教训都需要读者自己从头积累。换句话说，传统的语言书和算法之间存在不小的鸿沟。

基于上述问题，本书采取一种语言和算法相结合的方法，把内容分为如下 3 部分：

- 第 1 部分是语言篇（第 1~4 章），纯粹介绍语言，几乎不涉及算法，但逐步引入一些工程性的东西，如测试、断言、伪代码和迭代开发等。
- 第 2 部分是算法篇（第 5~8 章），在介绍算法的同时继续强化语言，补充了第 1 部

分没有涉及的语言特性，如位运算、动态内存管理等，并延续第一部分的风格，在需要时引入更多的思想和技巧。学习完前两部分的读者应当可以完成相当数量的练习题。

- 第3部分是竞赛篇（第9~11章），涉及竞赛中常用的其他知识点和技巧。和前两部分相比，第3部分涉及的内容更加广泛，其中还包括一些难以理解的“学术内容”，但其实这些才是算法的精髓。

本书最后有一个附录，介绍开发环境和开发方法，虽然它们和语言、算法的关系都不大，却往往能极大地影响选手的成绩。另外，本书讲解过程中所涉及的程序源代码可登录网站 <http://www.tup.tsinghua.edu.cn/> 进行下载。

致谢

在真正动笔之前，我邀请了一些对本书有兴趣的朋友一起探讨本书的框架和内容，并请他们撰写了一定数量的文字，他们是赖笠源（语言技巧、字符串）、曹正（数学）、邓凯宁（递归、状态空间搜索）、汪堃（数据结构基础）、王文一（算法设计）、胡昊（动态规划）。尽管这些文字本身并没有在最终的书稿中出现，但我从他们的努力中获得了很多启发。北京大学的杨斐瞳完成了本书中大部分插图的绘制，清华大学的杨锐和林芝恒对本书进行了文字校对、题目整理等工作，在此一并表示感谢。

在本书构思和初稿写作阶段，很多在一线教学的老师给我提出了有益的意见和建议，他们是绵阳南山中学的叶诗富老师、绵阳中学的曾贵胜老师、成都七中的张君亮老师、成都石室中学的文仲友老师、成都大弯中学的李植武老师、温州中学的舒春平老师，以及我的母校——重庆外国语学校的官兵老师等。

本书的习题主要来自 UVa 在线评测系统，感谢 Miguel Revilla 教授和 Carlos M. Casas Cuadrado 的大力支持。

最后，要特别感谢清华大学出版社的朱英彪编辑，与他的合作非常轻松、愉快。没有他的建议和鼓励，或许我无法鼓起勇气把“算法艺术与信息学竞赛”以丛书的全新面貌展现给读者。

刘汝佳

目 录

第1部分 语言篇

第1章 程序设计入门	1
1.1 算术表达式	1
1.2 变量及其输入	3
1.3 顺序结构程序设计	6
1.4 分支结构程序设计	9
1.5 注解与习题	13
1.5.1 C语言、C99、C11 及其他	13
1.5.2 数据类型与输入格式	14
1.5.3 习题	15
1.5.4 小结	16
第2章 循环结构程序设计	18
2.1 for 循环	18
2.2 while 循环和 do-while 循环	22
2.3 循环的代价	25
2.4 算法竞赛中的输入输出框架	27
2.5 注解与习题	34
2.5.1 习题	34
2.5.2 小结	36
第3章 数组和字符串	37
3.1 数组	37
3.2 字符数组	41
3.3 竞赛题目选讲	45
3.4 注解与习题	53
3.4.1 进位制与整数表示	54
3.4.2 思考题	55
3.4.3 黑盒测试和在线评测系统	55
3.4.4 例题一览与习题	56
3.4.5 小结	59
第4章 函数和递归	61
4.1 自定义函数和结构体	61

4.2	函数调用与参数传递	65
4.2.1	形参与实参	65
4.2.2	调用栈	66
4.2.3	用指针作参数	69
4.2.4	初学者易犯的错误	71
4.2.5	数组作为参数和返回值	71
4.2.6	把函数作为函数的参数	73
4.3	递归	74
4.3.1	递归定义	74
4.3.2	递归函数	75
4.3.3	C语言对递归的支持	75
4.3.4	段错误与栈溢出	77
4.4	竞赛题目选讲	79
4.5	注解与习题	92
4.5.1	头文件、副作用及其他	93
4.5.2	例题一览和习题	95
4.5.3	小结	99
第5章	C++与STL入门	100
5.1	从C到C++	100
5.1.1	C++版框架	101
5.1.2	引用	102
5.1.3	字符串	103
5.1.4	再谈结构体	105
5.1.5	模板	106
5.2	STL初步	108
5.2.1	排序与检索	108
5.2.2	不定长数组: <code>vector</code>	109
5.2.3	集合: <code>set</code>	112
5.2.4	映射: <code>map</code>	113
5.2.5	栈、队列与优先队列	115
5.2.6	测试STL	120
5.3	应用: 大整数类	123
5.3.1	大整数类 <code>BigInteger</code>	124
5.3.2	四则运算	125
5.3.3	比较运算符	126
5.4	竞赛题目举例	127

5.5 习题	134
--------------	-----

第2部分 基础篇

第6章 数据结构基础	139
6.1 再谈栈和队列	139
6.2 链表	143
6.3 树和二叉树	148
6.3.1 二叉树的编号	148
6.3.2 二叉树的层次遍历	150
6.3.3 二叉树的递归遍历	155
6.3.4 非二叉树	160
6.4 图	162
6.4.1 用DFS求连通块	162
6.4.2 用BFS求最短路	164
6.4.3 拓扑排序	167
6.4.4 欧拉回路	168
6.5 竞赛题目选讲	170
6.6 训练参考	175
第7章 暴力求解法	182
7.1 简单枚举	182
7.2 枚举排列	184
7.2.1 生成1~n的排列	184
7.2.2 生成可重集的排列	185
7.2.3 解答树	186
7.2.4 下一个排列	187
7.3 子集生成	188
7.3.1 增量构造法	188
7.3.2 位向量法	188
7.3.3 二进制法	189
7.4 回溯法	191
7.4.1 八皇后问题	191
7.4.2 其他应用举例	194
7.5 路径寻找问题	198
7.6 迭代加深搜索	206
7.7 竞赛题目选讲	209
7.8 训练参考	213