ACM国际大学生程序设计竞赛（ACM-ICPC）系列丛书

ACM-ICPC 基本算法

滕国文 李昊 编著

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-ICPC 基本算法

滕国文 李昊 编著

清华大学出版社
北京

内 容 简 介

本书简要介绍了 ACM-ICPC（ACM 国际大学生程序设计竞赛）、算法和算法设计的基础知识，重点讲解算法设计方法，给出了 ACM-ICPC 中常用的 10 种算法设计方法：求值法、递推法、递归法、枚举法、模拟法、分治法、贪心法、回溯法、构造法和动态规划法。本书针对每种程序设计方法，首先阐述该方法的基本思想，然后通过典型例题进行详细讲解，最后通过实战训练予以巩固和提高。

本书注重 ACM-ICPC 的基本算法，思想高度概括、例题深入浅出、实战耐人寻味。本书可作为 ACM 国际大学生程序设计竞赛和中学青少年信息学奥林匹克竞赛的指导书，也可作为 IT 技术人员和计算机编程爱好者的参考书。

图书在版编目（CIP）数据

ACM-ICPC 基本算法 / 滕国文，李昊编著. —北京：清华大学出版社，2018
（ACM 国际大学生程序设计竞赛（ACM-ICPC）系列丛书）
ISBN 978-7-302-50313-2

Ⅰ. ①A…　Ⅱ. ①滕…　②李…　Ⅲ. ①程序设计–算法–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（2018）第 114973 号

责任编辑： 袁勤勇　常建丽
封面设计： 傅瑞学
责任校对： 焦丽丽
责任印制： 丛怀宇

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×260mm　　**印　　张：** 14.5　　**字　　数：** 355 千字
版　　次： 2018 年 9 月第 1 版　　**印　　次：** 2018 年 9 月第 1 次印刷
定　　价： 39.00 元

产品编号：077040-01

前 言

ACM-ICPC 是国际计算机学会组织的国际大学生程序设计竞赛。这项赛事是大学生智力与计算机解题能力的竞赛，是大学生展示水平与才华的大舞台，是各高校计算机教育成果的直接体现，也是 IT 企业与世界顶尖计算机人才对话的最佳机会。因而，程序设计竞赛吸引了越来越多的高校参赛。

ACM-ICPC 中的“基本算法”用于指导学生分析问题和设计算法解决问题。学习常用基本算法设计方法，有助于理解算法设计的基本思想和科学原理，掌握算法设计的基本知识和基本技能，掌握算法设计中的计算思维和解题策略。

在本书各章的讨论中，首先介绍一种算法设计方法的基本思想，然后将计算机经典问题和算法设计方法很好地结合起来，运用该算法设计方法去解决这些问题，并给出 C 语言描述，最后通过实战训练予以巩固和提高，以达到融会贯通的效果。

本书的资料来源于吉林师范大学 ACM-ICPC 训练讲义，所选典型例题和实战训练题目分别来自吉林师范大学 ONLINE JUDGE（JLOJ），网址为 http://acm.jlnu.edu.cn；北京大学 ONLINE JUDGE（POJ），网址为 http:// poj.org；浙江大学 ONLINE JUDGE（ZOJ），网址为 http://acm.zju.edu.cn；杭州电子科技大学 ONLINE JUDGE（HDOJ），网址为 http://acm.hdu.edu.cn。

全书共 11 章。第 1 章简要介绍了 ACM-ICPC、算法、算法分析和优化的基础知识；第 2～11 章系统讲解了 10 种常用的算法设计方法，分别为求值法、递推法、递归法、枚举法、模拟法、分治法、贪心法、回溯法、构造法和动态规划法。

本书由吉林师范大学滕国文教授和李昊讲师撰写，从 2005 年开始，两位老师先后担任吉林师范大学 ACM-ICPC 代表队的主教练，开设了 ACM-ICPC 选修课，并成立了 ACM-ICPC 集训队，进行了十多年的教学尝试和实践训练。集训队主力队员施海勇、郭志鑫、李俊岐、李长彬和李婉琪参加了本书部分代码编写和程序调试工作，白文秀、曹宇、滕泰、于淼、温毓铭、王双印、李然、高巨、赵腾飞和米雪阳等参与了本书文稿的校对工作，作者在此一并致以诚挚的谢意！全书由滕国文教授统稿。

在本书的编写过程中，作者参阅并借鉴了国内外诸多同行的文章和著作，这里不一一列举、标明，在此向他们致以谢意！

由于作者水平有限，加之学科理论与技术发展日新月异，对于书中疏漏与不妥之处，恳请广大读者批评指正。

作　者

2018 年 1 月

目 录

第1章 ACM与算法概述

1.1 ACM-ICPC简介

ACM国际大学生程序设计竞赛（ACM International Collegiate Programming Contest，ACM-ICPC）由美国计算机协会（Association for Computing Machinery）主办，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，有“程序设计的奥林匹克”之美称。其目的是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。该项竞赛云集了计算机界的“精英”，引起国际各知名大学的重视，受到全世界各著名计算机公司的高度关注，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-ICPC的宗旨是培养参赛选手的创造力、团队合作精神以及他们在软件程序开发过程中的创新意识，同时也检测选手们在压力下进行开发活动的能力。可以说，ACM国际大学生程序设计竞赛是参赛选手展示计算机才华的广阔舞台，是著名大学计算机教育成果的直接体现，是信息企业与世界顶尖计算机人才对话的最好机会。

1.1.1 历史

ACM-ICPC于1970年，在美国得克萨斯A & M大学举办了首届比赛，当时的主办方是UPE计算机科学荣誉协会Alpha分会（the Alpha Chapter of the UPE Computer Science Honor Society）。作为一种全新的发现和培养计算机科学顶尖学生的方式，竞赛很快得到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，并演变成为目前的一年一届的国际性比赛。

最初几届比赛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，特别是自1997年IBM开始赞助赛事之后，赛事规模增长迅速。1997年，共有来自560所大学的840支队伍参加比赛。到了2004年，这一数字迅速增加到840所大学的4109支队伍，并以每年10%～20%的速度增长。1980年，ACM将竞赛的总部设在位于美国得克萨斯州的贝勒大学。在赛事的早期，冠军多为美国和加拿大的大学。进入20世纪90年代后期，俄罗斯和其他一些东欧国家的大学连夺数次冠军。来自中国的上海交通大学代表队则在2002年美国夏威夷第26届、2005年上海举行的第29届和2010年中国哈尔滨的第34届全球总决赛上三夺世界冠军；2011年，在美国奥兰多举行的第35届全球总决赛上，浙江大学获得世界冠军。这是到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的亚欧对抗的局面。

1.1.2 比赛规则

ACM-ICPC 以团队的形式代表各学校参赛，每队由 3 名队员组成。每位队员必须是入校 5 年内的在校大学生，最多可以参加 2 次全球总决赛和 4 次区域选拔赛。比赛期间，每队使用 1 台计算机，需要在 5 小时内使用 C、C++、Pascal 或 Java 中的一种语言编写程序解决 8 或 10 个问题（通常是区域选拔赛 8 题，全球总决赛 10 题）。程序完成之后提交给在线评测系统去运行，运行的结果为正确或错误两种并及时通知参赛队。每道试题用时将从竞赛开始到试题解答被判定为正确为止，其间每次提交运行结果被判错误将被加罚 20 分钟时间，未正确解答的试题不计时。每队在正确完成一题后，组织者将在其位置上升起一只代表该题颜色的气球。每道题目的第一支解决队伍还会额外获得一个 FIRST PROBLEM SOLVED 气球。

竞赛结束后，参赛各队以解出问题的多少进行排名，规则如下：

（1）正确解决题目数较多的参赛队排名在前。

（2）对于正确解决题目数相同的参赛队，总罚时少的参赛队排名在前。

罚时的计算方法如下：

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，这一题目的罚时=从比赛开始到正确解决这道题目经过的分钟数+正确解决之前错误的提交次数×20 分钟。例如，A、B 两队都正确完成两道题目，其中 A 队提交这两题的时间分别是比赛开始后 1:00 和 2:45，B 队为 1:20 和 2:00，但 B 队有一题提交了 2 次。这样，A 队的总用时为 1:00+2:45=3:45，而 B 队为 1:20+2:00+0:20=3:40，所以 B 队以总用时少而获胜。

1.1.3 区域和全球决赛

与其他计算机程序竞赛（如国际信息学奥林匹克竞赛，IOI）相比，ACM-ICPC 的特点在于其题量大，每队需要 5 小时内完成 8 道题目，甚至更多。另外，一支队伍 3 名队员只有 1 台计算机，使得时间显得更为紧张。因此，除了扎实的专业水平，良好的团队协作和心理素质同样是获胜的关键。

该项竞赛分区域预赛和全球总决赛两个阶段进行。各预赛区第一名自动获得参加全球总决赛的资格。决赛安排在每年的 3—4 月举行，而区域预赛一般安排在上一年的 9—12 月举行。一个大学可以有多支队伍参加区域预赛，但只能有一支队伍参加全球总决赛。

全球总决赛第一名将获得奖杯一座。另外，成绩靠前的参赛队伍也将获得金、银或铜牌。而解题数在中等以下的队伍会得到确认，但不会进行排名。

1.2 算法与问题求解

用计算机解决实际问题时，最终要编写程序。根据著名的计算机科学家尼克拉斯·沃思（Niklaus Wirth）提出的著名公式：

算法＋数据结构＝程序

可以看出，要编写程序，必须确定算法和数据结构。

1.2.1　算法的定义

算法就是为了求解问题而给出的指令序列，可以理解为由基本运算及规定的运算顺序构成的完整的解题步骤，而程序是算法的一种实现。计算机按照程序逐步执行算法，实现对问题的求解。简单说，算法可以看成是按照要求设计好的有限的确切的计算序列，并且这样的步骤和序列可以解决某个（类）问题。

算法设计的重点是把人类找到的求解问题的方法、步骤，以过程化、形式化、机械化的形式表示出来，以便让计算机执行。

对于一个问题，如果可以通过一个计算机程序在有限的存储空间内运行有限的时间而得到正确的结果，则称这个问题是算法可解的，但算法不等于程序，也不等于计算方法。当然，程序也可以作为算法的一种描述，但程序通常还需要考虑很多与方法和分析无关的细节问题，这是因为在编写程序时要受到计算机系统环境的限制。通常，程序的编制不可能优于算法的设计。

算法是一个十分古老的研究课题，然而计算机的出现为这个课题注入了青春和活力，使算法的设计和分析成为计算机科学中的研究热点之一。

1967 年，D.E.Knuth 指出：“算法是贯穿在所有计算机程序设计中的一个基本概念。”所以，算法被誉为计算机学科的灵魂。

数学大师吴文俊指出：“我国传统数学在从问题出发以解决问题为主旨的发展过程中，建立了以构造性与机械化为其特色的算法体系，这与西方数学以欧几里得《几何原本》为代表的所谓公理化演绎体系正好遥遥相对。……肇始于我国的这种机械化体系，在经过明代以来近几百年的相对消沉后，由于计算机的出现，已越来越为数学家所认识与重视，势将重新登上历史舞台。”吴文俊创立的几何定理的机器证明方法（世称吴方法）用现代的算法理论，使得中国古代传统算法发扬光大，受到国家的高度关注，也享有很高的国际声誉。

1.2.2　问题求解

用计算机解决实际问题，就是在计算机中建立一个解决问题的模型。在这个模型中，计算机内部的数据表示了需要被处理的实际对象，包括其内在的性质和关系、处理这些数据的程序、模拟对象领域中的求解过程。通过解释计算机程序的运行结果，便得到了实际问题的解。下面给出用计算机求解问题的一般步骤。

1. 问题分析

这个阶段的任务是弄清题目提供的已知信息和所要解决的问题。完整地理解和描述问题是解决问题的关键。要做到这一点，必须注意以下问题：是否明白在未经加工的原始表达中用的术语的准确定义？题目提供了哪些已知信息？还可以得到哪些潜在的信息？题目中做了哪些假定？题目要求得到什么结果？等等。针对每个具体问题，必须认真审查问题的有关描述，深入分析，以加深对问题的准确理解。

2. 数学模型的建立

用计算机解决实际问题，必须建立正确的数学模型。因为在现实问题前，计算机是无能为力的。对一个实际问题建立数学模型，可以考虑这样两个基本问题：最适合此问题的数学模型是什么？是否有已经解决了的类似问题可以借鉴？

建立数学模型是最关键且较困难的一步，涉及 4 个世界和三级抽象。4 个世界分别是：现实世界（客观世界）、信息世界（概念世界）、数据世界、计算机世界；三级抽象分别是：现实世界到信息世界的抽象，建立信息模型或概念模型；信息世界到数据世界的抽象，转化信息数据建立数据模型；数据世界到计算机世界的抽象，建立存储模型在计算机中实现。

3. 算法设计

算法设计是指设计求解某一特定类型问题的一系列步骤，并且这些步骤是可以通过计算机的基本操作来实现的。算法设计要同时结合数据结构的设计。简单说，数据结构的设计就是选取存储方式，因为不同的数据结构的设计将导致算法的差异很大。算法的设计与模型的选择更是密切相关的，但同一模型仍然可以有不同的算法，而且它们的有效性可能有相当大的差距。

算法设计方法也称算法设计技术，或算法设计策略，是设计算法的一般性方法，可用于解决不同计算领域的多种问题。虽然设计算法，尤其是设计出好的算法是一项非常困难的工作，但是设计算法也不是没有方法可循，人们经过几十年的工作，总结和积累了许多行之有效的方法，了解和掌握这些方法会给我们解决问题提供一些思路。本书讨论的算法设计方法已经被证明是对算法设计非常实用的通用技术，包括求值法、递推法、递归法、枚举法、模拟法、分治法、贪心法、回溯法、构造法和动态规划法等。这些算法设计方法构成了一组强有力的工具，可用于大量实际问题的求解。

4. 算法表示

对于复杂的问题，确定算法后可以选择一种算法描述方法来准确表示算法。算法的描述方式有很多，如传统流程图、盒图、PAD 图、伪码和高级语言等。其中，高级语言是最理想的描述算法的方法，因此本书选择 C 语言来表示算法。

5. 算法分析

算法分析的目的，首先是为了对算法的某些特定输入，估算该算法所需的内存空间和运行时间，其次是为了建立衡量算法优劣的标准，用以比较同一类问题的不同算法。一般来说，一个好的算法首先应该是比同类算法的时间效率高，算法的时间效率用时间复杂度来度量。

6. 算法实现

算法实现是指编码，也就是平常说的编程序，即将算法设计“转译”成某种计算机语言的表述形式，才能够在计算机上执行。编码的目的是使用选定的程序设计语言把算法描述翻译成为用该语言编写的源程序（或源代码）。源程序应该正确可靠、简明清晰，而且具有较高的效率。

在把算法转变为程序的过程中，虽然现代编译器提供了代码优化功能，但是仍然需要一些技巧，如在循环之外计算循环中的不变式、合并公共子表达式等。

7. 程序调试

程序调试也称算法测试，其任务首先是发现和排除在前几个阶段中产生的错误，经测试通过的程序才可投入运行，在运行过程中还可能发现隐含的错误和问题，因此，必须在使用中不断地维护和完善。

算法测试的实质是对算法应完成任务的实验证实，同时确定算法的使用范围。测试方法一般有两种：白盒测试，对算法的各个分支进行测试；黑盒测试，检验对给定的输入是否有指定的输出。

8. 整理结果、编制文档

整理结果时，要对计算结果进行分析，看其是否符合实际问题的要求，如果符合，问题得到解决，可以结束；如果不符合，说明前面的步骤一定存在问题，必须返回，从头开始逐步检查，找出错误并重新设计，这个循环过程也可能重复多次。

编制文档的目的是让别人理解你编写的算法。首先要把代码编写清楚，代码本身就是文档，同时还有代码的注释。另外还包括算法的流程图，自顶向下各研制阶段的相关记录，算法的正确性证明（论述），算法测试过程、结果，对输入输出的要求及格式的详细描述等。

1.3　算法的特性

1.3.1　算法的要素

算法由操作、控制结构和数据结构三要素组成。

1. 操作

算法实现平台尽管有许多种类，它们的函数库、类库也有较大差异，但是必须具备的最基本的操作功能是相同的。这些操作包括：

算术运算：加法、减法、乘法、除法等运算。

关系比较：大于、小于、等于、不等于等运算。

逻辑运算：与、或、非等运算。

数据传送：输入、输出、赋值等操作。

2. 控制结构

一个算法功能的实现不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。算法的控制结构给出了算法的基本框架，它不仅决定了算法中各操作的执行顺序，而且也直接反映了算法的设计是否符合结构化原则。

算法的基本控制结构有下列 3 种。

（1）顺序结构：顺序结构是程序设计中最简单、最常用的基本结构。在该结构中，各

操作块按照出现的先后顺序依次执行。它是任何程序的主体基本结构，即使在选择结构或循环结构中，也常以顺序结构作为其子结构。

（2）选择结构：又称为分支结构，是指程序依据条件所列出表达式的结果来决定执行多个分支中的哪一个分支，进而改变程序执行的流程。依据条件选择分支的结构称为选择结构。

（3）循环结构：某一类问题可能需要重复多次执行完全一样的计算和处理方法，而每次使用的数据都按照一定的规律在改变。这种可能重复执行多次的结构称为循环结构，又称重复结构。

3. 数据结构

算法操作的对象是数据，数据间的逻辑关系、数据的存储方式及处理方式就是数据结构。它与算法设计是紧密相关的。

有了计算机的帮助，许多过去靠人工无法计算的大量复杂问题就有了解决的希望。不过，使用计算机进行计算，首先要解决的是如何将被处理的对象存储到计算机中，也就是要选择适当的数据结构。

1.3.2 算法的基本特性

算法应具有以下 5 个重要特性。

（1）输入：一个算法应有 0 个或多个外部量作为算法的输入。有些输入量需要在算法执行过程中输入，而有些算法表面上可以没有输入，实际上已被嵌入算法之中。

（2）输出：一个算法应产生一个或多个量作为输出。它是一组与输入有确定关系的量值，是算法进行信息加工后得到的结果。

（3）确定性：算法中的每条指令必须有确切的含义，无二义性，即每种情况下应执行的操作在算法中都有确切的规定，使算法的执行者或阅读者都能明确其含义及如何执行。在任何条件下，对于相同的输入，只能得到相同的输出。

（4）有穷性：指算法必须能在执行有限步骤后、有限的时间内终止，即每条指令的执行次数和执行时间必须是有限的。

（5）可行性：算法描述的操作可以通过已经实现的基本操作执行有限次来实现。就是指算法的每个步骤，计算机都能执行。计算机能执行的动作是预先设计好的，一旦出厂就不会改变。所以，设计算法时，应考虑每个步骤必须能用计算机所能执行的操作命令实现。

综上所述，算法是一组严谨定义运算顺序的规则，并且每个规则都是有效的、明确的，此顺序将在有限的次数下终止。

1.4 算法的描述

算法设计者在构思和设计了一个算法之后，必须清楚准确地将所设计的求解步骤记录下来，即算法描述。常用的算法描述方法有：自然语言、传统流程图、N-S 图、PAD 图、伪代码和高级语言等。本书使用高级语言中的 C 语言来描述算法。

下面首先给出 3 种基本控制结构的描述，然后详细列出 C 算法描述的约定。

1.4.1　基本控制结构的描述

计算机科学家已经证明只使用 3 种基本控制结构就可以构建解决任何复杂问题的算法。这 3 种基本控制结构是：顺序结构、选择结构和循环结构。

为了更好地领会和理解这 3 种基本控制结构，下面对照给出 C 语言与 N-S 图两种描述方法，并作如下约定：S（Statement），S1，S2 代表语句；E（Expression），E1,E2,E3 代表表达式；T（True）代表逻辑“真”（成立），F（False）代表逻辑“假”（不成立）。

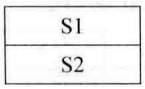
1. 顺序结构

① C 语言描述：

```
S1
S2
```

即 S1 语句在前，S2 语句在后。

② N-S 图描述：



顺序结构是指程序的执行次序与程序的书写次序一致，即先执行 S1，后执行 S2。

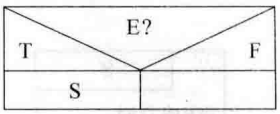
2. 选择结构

1）简单选择结构

① C 语言描述：

```
if( E )
    S
```

② N-S 图描述：



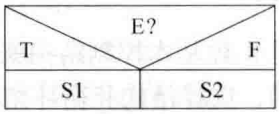
简单选择结构执行时，先判断表达式 E，若为非零（成立），则执行 S；否则什么都不做。

2）一般选择结构

① C 语言描述：

```
if ( E )
    S1
else
    S2
```

② N-S 图描述：



一般选择结构执行时，先判断表达式 E，若为非零（成立），则执行 S1；否则执行 S2。S1 与 S2 只执行其一。

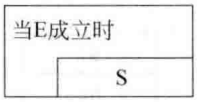
3. 循环结构

1）while 循环

① C 语言描述：

```
while ( E )
 S
```

② N-S 图描述：



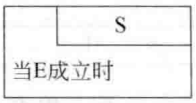
while 循环执行过程：首先判断表达式 E，若为非零（成立），则执行语句 S；执行语句 S 后，再返回表达式 E 的判断，如果仍为非零（成立），再次执行语句 S；如此反复，直到某次表达式 E 为零（不成立）为止，结束该循环。

2）do-while 循环

① C 语言描述：

```
do
   S
while ( E ) ;
```

② N-S 图描述：



do-while 循环执行过程：首先执行一次语句 S，然后判断表达式 E，若为非零（成立）时，则再次执行语句 S；执行语句 S 后，再返回表达式 E 的判断，如果仍为非零（成立），再次执行语句 S；如此反复，直到某次表达式 E 为零（不成立）为止，结束该循环。

3）for 循环

① C 语言描述：

```
for (E1; E2; E3)
   S
```