



中国计算机学会信息学奥林匹克系列丛书



信息学奥林匹克竞赛

Olympiad in Informatics

国际国内分类试题精解 (2003-2004)

上册

主 编 中国计算机学会信息学奥林匹克科学委员会

本册编著 吴文虎 王建德

清华大学出版社





www.pearsoned.com.au



UNIT 10

City council for Information

Unit 10: City council for Information

Unit 10

Unit 10: City council for Information

Unit 10: City council for Information



中国计算机学会信息学奥林匹克系列丛书

No. 4

信息学奥林匹克竞赛
Olympiad in Informatics
国际国内分类试题精解(2003—2004)
上册

主编 中国计算机学会信息学奥林匹克科学委员会

本册编著 吴文虎 王建德

清华大学出版社

北 京

内 容 简 介

《中国计算机学会信息学奥林匹克系列丛书》由中国计算机学会信息学奥林匹克科学委员会主编,由全国著名专家学者精心编著而成。

本书收录了2003年至2004年国际国内信息学奥林匹克竞赛的大部分试题。全书对试题进行了类型归纳,并分上、下两册出版。上册包括基础类试题、数据结构类试题、搜索类试题和动态程序设计类试题。下册包括计算几何类试题和构造类试题。全书对每种类型试题作了简要的介绍,所有的试题都给出了具体的算法分析和相应的源代码。

本书既适合教师辅导学生使用,也适合参加信息学奥林匹克竞赛的学生自学,同时也是大专院校的计算机爱好者学习编程的优秀参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

信息学奥林匹克竞赛.上册:国际国内分类试题精解:2003~2004/吴文虎,王建德编著.一北京:清华大学出版社,2005.8

(中国计算机学会信息学奥林匹克系列丛书)

ISBN 7-302-11636-9

I. 信… II. ①吴… ②王… III. 计算机课—中小学—解题 IV. G634.675

中国版本图书馆CIP数据核字(2005)第093244号

出 版 者:清华大学出版社

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

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:孙中悦

印 装 者:北京市清华园胶印厂

发 行 者:新华书店总店北京发行所

开 本:185×230 印张:13.5 字数:265千字

版 次:2005年8月第1版 2005年8月第1次印刷

书 号:ISBN 7-302-11636-9/TP·7601

印 数:1~5000

定 价:19.80元

PREFACE

序 言

信息技术对人类社会的发展产生着深远的影响,已成为新世纪的一个标志。作为人类集体智慧的结晶,信息技术已成为一种时代文化。“计算机的普及要从娃娃抓起”成为“科教兴国”的一项重要内容。

一个国家、一个民族要立足于世界先进民族之林,关键在于拥有高素质的人才。综合国力的竞争,说到底人才的竞争,培养和造就一大批优秀信息技术人才是当务之急。信息时代,信息技术已成为现代科学技术的基础核心,成为人类的“通用智力工具”,在青少年中普及信息技术教育具有重要和深远的意义。

中国计算机学会从1984年起,就组织青少年参加信息学奥林匹克竞赛。二十余年,学会通过组织竞赛推动信息技术普及,促进青少年掌握信息技术知识,并提高他们的逻辑思维和解决问题的能力。为了培养和造就更多高素质的信息技术人才,中国计算机学会特别推出一套信息学奥林匹克系列指导丛书。这套丛书从基础知识开始,重点培养学生的创新思维方法和编程能力。本书的编著者大多是多年从事信息技术普及教学和信息学奥林匹克竞赛的指导者,有丰富的教学经验,他们编写的信息学奥林匹克系列丛书受到了全国青少年的喜爱,他们指导的学生曾多次在国际上获得金奖,为培养青少年的信息素养提供了很好的精神食粮。编著这套丛书的目的旨在培养学生逻辑思维、创新能力和全面提高青少年素质方面提供帮助。

该丛书的编写是开放式的,凡有志于向青少年普及信息技术的科技工作者和教育工作者都可以加入到这个行列中。

中国计算机学会(代理)理事长
中国工程院院士



2003年11月

FOREWORD

前言

著名的计算机科学家、图灵奖的获得者、美国斯坦福大学教授 G. 伏赛斯曾预言：计算机科学将是继自然语言、数学之后，成为第三个对人的一生都有重大用途的“通用智力工具”。20 世纪末，以计算机为龙头的信息技术风起云涌般的发展势头，将世界带入了一个快速发展的新时代。竞争中的领跑者们意识到要不失时机地将有关信息科学知识与应用能力，尽快地纳入到青少年的知识结构中。信息科技应该成为跨世纪人才迈向信息时代的入场券。就是在这样的背景下，1987 年，保加利亚的 Sendov 教授在联合国教科文组织第 24 届全体会议上，倡议举办国际信息学奥林匹克竞赛(IOI)。首届竞赛于 1989 年 5 月在保加利亚的布拉维茨举行，当年有 13 个国家的 46 名选手参赛。此后每年举办一届，到 2004 年已举办了 16 届，有 70 多个国家和地区派队参加，可以说盛况空前。

我国自 1984 年起，开始举办全国青少年信息学奥林匹克竞赛(NOI)，到 2004 年已举办了 21 届。NOI 以及全国青少年信息学奥林匹克联赛(NOIP)等系列活动旨在向青少年普及计算机科学知识，为学校的信息技术教育提供动力和新的思路，给那些有才华的学生提供相互交流和学习的机会，通过竞赛和相关的活动培养和选拔优秀的计算机人才。

信息学奥林匹克竞赛属于智力与应用计算机解题能力的比赛。题目有相当的难度，而且有越来越难的趋势。解好这类题目，需要有相当强的数、理、化、文、史、地的基础，具备很强的全面素质与综合能力。首先是读题能力，英语和语文好的当然占先；之后是观察和分析问题的能力，一下就能切入主题抓住本质者占先；再就是将实际问题进行抽象并转化为数学模型的能力，这就要求数学功底要强，最好超强；接着是灵活运用各种算法的能力，有时还得自己针对题目设计新的算法；想好算法之后，要在某种语言环境下，熟练地编写程序，并

将其快速无误地调试通过；最后一条是根据题目的要求和边界条件的估计，自己设计测试数据，检查自己的算法及其实现是否正确，是否完备。

奥林匹克学科竞赛活动是一种特殊的训练，特殊在学习理念和模式上。这种活动属于智力活动。智力活动的核心是思维，思维是人脑的主要功能。通过学习电脑帮助开发人脑是信息学奥林匹克竞赛活动的一个特点。这种智力训练纸上谈兵不行，一定要营造良好的实践环境，引导选手利用计算机这样一个智力工具，自己寻找问题、分析问题。教育心理学家认为，培养创造意识要在学习中倡导发现法，使学习活动尽可能带有探索性，让学习者始终处于探索、追求、刻意求新、力求完美的精神状态之下。这种探索性学习的心理基础应该是“自立、自信、自尊、自强”，内在动力是强烈的兴趣。恰恰信息学竞赛活动能以其难和新，激发兴趣。竞赛中有很多题目尚无固定解法，极具挑战性，能够吸引学生自主进行研究性学习，在这一过程中学会如何透过现象看清本质，如何在纷繁的矛盾中抓住主要矛盾，切入主题，如何运用所学知识对问题进行抽象、形式化和模型化，如何通过编程将问题加以解决。在这一过程中，选手们要去钻研很多可能在大学阶段才能学到的知识，比如数据结构、组合数学、人工智能搜索算法、图论与集合论等。这个学习任务不是老师强加给选手的，而是研究性学习当中他们自己必然会遇到的，兴趣使然，再难他们也愿意去学。这种研究性学习的最大好处是能使选手有效地提高科学思维能力。我以为科学思维能力的提高是成就事业最重要的一个因素。

就我个人教学中的感受而言，算法的确是艺术。艺术与科学本来就是孪生姊妹，算法属于数学文化范畴，数学的美在算法中得到了充分的体现，特别是当今数学已经进入了新的机器时代，利用计算机求解问题，需要充分开动脑筋，解题过程本身就是一个精益求精追求完美的过程。在这样一个过程中，编程者在付出艰辛之后，会有一种获得成功的愉悦。正像一些数学大师说的：数学是理性的艺术，是创造性的艺术。在编程解题中，通过理性思维和理性实践，选手们感受到了思维艺术之美。因此，我们许多参加过 NOI 和 IOI 大赛的选手，当问起他们当年的拼搏是否很苦时，他们会说苦中有乐，苦中有甜，这可能就是感受到了这种思维艺术魅力的缘故吧。

吴文虎

· 2005 年 8 月

CONTENTS

目 录

第 1 章	2003 年至 2004 年信息学奥林匹克活动概况	1
1.1	近两年信息学奥林匹克竞赛试题的特点	1
1.2	近两年信息学奥林匹克活动的启示	4
第 2 章	基础类试题	11
2.1	不高兴的津津	11
2.2	津津的储蓄计划	13
2.3	乒乓球	15
2.4	麦森数	18
第 3 章	数据结构类试题	23
3.1	花生采摘	24
3.2	火星人	28
3.3	合并果子	31
3.4	FBI 树	36
3.5	神经网络	39
3.6	文本编辑器	43
3.7	出纳员	53
3.8	路径维护	60
3.9	草莓	66
3.10	数据生成器	78

第 4 章 搜索类试题	87
4.1 雕刻家	89
4.2 Empodia	92
4.3 侦探推理	96
4.4 虫食算	102
4.5 代码转换	111
4.6 木棒游戏	120
4.7 机器人	127
4.8 栈	135
4.9 传染病控制	138
4.10 智破连环阵	142
第 5 章 动态程序设计类试题	151
5.1 数字游戏	153
5.2 加分二叉树	156
5.3 合唱队形	159
5.4 农夫	162
5.5 小 H 的小屋	168
5.6 曼哈顿	173
5.7 网络改造	187

2003年至2004年信息学奥林匹克活动概况

2003年至2004年,在中国科协的领导下,中国计算机学会组织了两届全国青少年信息学奥林匹克联赛(简称 NOIP)、全国青少年信息学奥林匹克竞赛(简称 NOI)、国际信息学奥林匹克竞赛中国队选拔赛(简称 CTSC,2003 年的国际信息学奥林匹克竞赛中国队选拔赛因“非典”的缘故被临时取消),同时连续两年派队参加国际信息学奥林匹克竞赛(简称 IOI),以选拔在计算机科学有特长的优秀学生,推动信息科技知识的普及。

这两年的 7 次竞赛活动共推出了 46 道试题,这些试题一经网上公布后,立即吸引了成千上万热心信息学活动的师生参与讨论和研究;许多大学在组织相关课程教学或竞赛培训活动,也纷纷从网上下载试题,作为教学实例或练习题。所有关注这两年信息学活动的大中学生期盼试题的标准解答,想从中推测今后信息学活动的走势,获取提升自己编程能力的有用信息。正是为了顺应大家的这些需求,我们着手编写了这本书。

1.1 近两年信息学奥林匹克竞赛试题的特点

2003 年至 2004 年信息学竞赛的试题和往年一样,恪守竞赛大纲的要求,题意描述清晰,无二义性。难度区分与选拔目标紧扣,既与选手的知识基础和接受能力相适应,又鼓励选手在力所能及的范围内“跳一跳”,探索并掌握新知。

当然,这两年竞赛的试题也呈现出一些“与时俱进”的新特点。

1. 凸现信息学知识和数学知识整合的趋势

为了考核学生的数学能力,激发学生的创造力,这两年的信息学竞赛引入了一些需要数学建模的试题。例如,2003 年 NOIP 的《栈》要求计算 Catalan 数的组合公式;2004 年 NOIP 的《虫食算》涉及了线性代数高斯消元法;2004 年 CTSC 的《数字搜索》的解题过程

需要引入离散数学的自动机原理;《石器时代》要求对众数信息进行动态维护等。尤其是计算几何类的试题激增,2003年的NOI出现了1道,2004年的NOI、IOI和CTSC出现了5道。这些几何类的试题要求选手将抽象的几何问题、直观的几何图形与灵活的编程技术有机地联系起来,将抽象思维与形象思维有效地结合起来,准确地判断、分析和解决“形”的问题。尽管目前的几何问题一般为二维平面内点、直线、矩形或凸多边形的计算,还未涉及三维空间或曲线,但已经出现了向量计算,深度和难度在逐年增加,解法也呈现出多样化的趋势。

这一切说明信息学与数学的依赖关系日益凸现,以至于出现了一个有趣的现象:数学素质好的人虽然不一定会编程,但意识到要学会编程的人愈来愈多;编程解题能力强的人大都有数学的潜质和爱好,他们中愈来愈多的人也希望深造数学。两门学科的交融和整合是信息学奥林匹克活动发展的一个大趋势。

2. 强调基础知识的灵活应用

这两年,每年面向普及的NOIP竞赛都含有相当数量的基础题,用以考核选手的学科知识和程序设计的基本知识掌握得如何,是否会编程解决问题。

特别要提醒读者的是,在一些看似容易的基础题中潜伏着“陷阱”。例如,2003年NOIP的《传染病扩展》,极易诱导选手错误地使用贪心法或动态程序设计。实际上,该问题不具备最优子结构的特征,正确的算法是回溯搜索。

基础题不仅出现在NOIP竞赛中,而且也出现在高层次的NOI、CTSC和IOI的竞赛中,但对选手灵活应用基础知识的要求远高于普及性的NOIP竞赛。例如,2003年IOI的《路径维护》在动态生成无向图的过程中计算最小生成树。2003年NOI的《文本编辑器》要求设计多样的数据结构。例如,线性表、非线性的树和图。《数据生成器》必须综合使用前序遍历和后序遍历。

3. 突出对选手综合能力的要求

这两年竞赛的试题并不一定随知识的分类而发生变化,有时几乎找不到一个单一的经典算法,也找不到一个纯粹的数据结构问题,开始呈现出综合性的发展趋势。例如,2003年NOI的《草莓》涉及了链、环、树和完全无向图的计算,要求计算权和最小的一个连通分支。解题的方法既有动态程序设计方法,又有二分法和贪心法。

多种算法和多种数据结构的要求交织在同一题中,关键是看选手从哪个角度去分析,也就是说能不能综合所学的知识,运用自如地解决问题。选手的综合素质愈高,获胜的几率就愈大。

这里需要说明的是,本书按照知识分类排列各章节,仅是一种粗略的编排。实际上,许多试题涉及的知识是多方面的,我们只不过是根据其中起主导作用的因素进行归类。

4. 鼓励选手探索“未知”、创新思维

这两年信息学奥林匹克竞赛的试题,在题型上发生了如下一些变化:

- 树形结构类的问题增多。例如,2003年NOI的《卫星探测》通过二分查找构造一棵隐形的二叉树;《数据生成器》使用了容易被忽视的后序遍历;IOI的《路径维护》要求在往无向图添边的过程中计算最小生成树;2004年NOI的《曼哈顿》引入了线段树的算法等。这就要求选手能站到一个新的高度,探索非线性结构问题中的“未知”,展开生动活泼的策略创造。
- 近似算法类的试题增加。2003年的NOI和IOI共出现了6道近似算法类的试题;2004年的IOI和CTSC也出现了6道近似算法类的试题。由于这些例题视程序逼近最优解和最优效率的程度给分,因此拓展了选手思维的空间。由于近似算法类的选题空间比较大,因此题目的难度一般也比较大。近似算法类的试题按提交的方式分为交互式试题和答案提交型试题两种。

交互式试题:此种提交方式一般由测试方提供交互函数。要求选手巧妙“试探”,通过尽可能少地调用交互函数来计算出解。2003年的NOI、IOI和2004年的IOI、CTSC各出现了两道交互式试题。

答案提交型试题:此种提交方式测试数据“透明”,供选手观察分析,并且测试方提供了测试工具,让选手直接观察自己程序的运行结果,反思算法,调试纠错。2003年的NOI出现了一道答案提交型试题,2004年的IOI和CTSC出现了两道答案提交型试题。

这两类试题一般是计算可行方案或近似解,根据选手的解与标准答案的接近程度判分。竞赛方推出这些新题型的目的是让选手从推导最佳解的逻辑演绎中解放出来,通过多样化的解题途径激发其探索精神、创新意识和分析解决问题的能力。

- 可“一题多解”和开放性的试题增多。例如,2003年IOI的《逆向输出》就是一道没有固定模式和经典算法可套用、需要量身定制合适算法的试题。我们将在“第7章 构造类试题”中解析这些开放性的试题。在这两年的竞赛中,可“一题多解”的试题也不少。例如,2003年NOIP的《栈》既可以通过回溯法计算,也可以通过组合数学的Catalan数计算;2004年NOIP的《虫食算》既可用枚举+递推的算法,也可用高斯消元法+进位枚举的算法。

上述题型的变化,使得算法知识的不确定性和不稳定性增加,以至于选手经常面对算法多元化或算法陌生的试题,面对不知如何处置的情境。可以预见,在今后的信息学奥林匹克竞赛中,模式化的试题会越来越少,益智型试题会越来越多,试题的数学味会越来越浓,对选手的智能和数学功底的要求会越来越高。这就要求我们的选手既要娴熟于基本的数据结构和算法,又能够应对不同的问题情境,独立地设计数据结构与算法;既要有归纳能力,能够通过求同思维找出试题中蕴含的规律,设计解析式,更要有创造能力,

不拘泥于一个模式,能够依据问题线索发散求异思维,从不同角度寻求解决问题的多种可能途径。

“创造能力=知识积累+发散思维”。新题型拓宽了选手的思维空间,为发散思维和创造能力的培养注入了活力。

1.2 近两年信息学奥林匹克活动的启示

根据这两年信息学竞赛试题的变化,应该对培训活动的基本要求和目标重新定位:使学生正确地理解问题,深入问题的空间进行理性思维和理性实践,创造性地应对没有遇到过的挑战。

就实现这一目标和要求,提几点建议:

1. 注意处理好几个关系

(1) 处理好“自信心”和“自知之明”的关系。按照多元智力理论,自知—自省智力是指建构正确的自我知觉能力。在学习程序设计的初始阶段,入门的关键在于克服困难的勇气和自信心;而在进入了成熟期以后,想应用程序设计知识破解难题大题,主要取决于自省力和坚忍不拔的毅力。科学精神的核心是怀疑和批评。这种态度既对他人又对自己。“知人者智,自知者明”,“知常曰明”。自信心和自知之明是对立的统一。一个人没有自信心,则一事无成,但自信心倘若不是建立在自知之明的基础上,则会变成盲目和自负,不可能取得持续的成功。因此,我们在日常训练中,不能用“看书”取代“上机实践”,不能用“读懂”等同“会做”。经常要反思:为什么别人会这么想问题;如果由自己当场独立解题的话,会得到怎样的结果。纸上得来终觉浅,绝知此事要躬行。看懂了不稀奇,会做有灵气,能讲清原理、提出质疑、拓延思路、获取新知才算真正的成功。

(2) 处理好“夯实基础”和“破解难题”的关系。2004年中国队在IOI中获得团体第二的佳绩。我们认为,这和NOI与CTSC的选拔机制有一定的联系。NOI中“难”的试题选拔出有扎实功底的选手;而CTSC的“大题、难题”又在NOI的基础上遴选出有毅力和能力应对复杂挑战的选手。CTSC试题,无论是设计数据结构和算法的难度还是编程复杂度,都是历届竞赛所未有的,不仅需要选手扎实的功底、缜密和敏捷的思维,而且考验选手心理的承受力和意志力。而这一切,离不开选手平日的基础性训练。

编程解题能力是在长期的上机实践中积累起来的。我们平日里千万不要有“太易了不屑做、太难了不愿做”的主观随意性。长此以往,迟早会落得个“小题中题做不对、大题难题不会做”。应该调整好自己的心态,切忌浮躁和急功近利。既要通过认真做“小题中题”夯实基础,又要不畏“大题、难题”,自觉培养内省智力,磨砺攻克难关的能力和意志力。在破解“大题、难题”的过程中,哪怕出现一两次失误,也是砥砺之石,可从中汲取有益的经验教训,“不是一番寒彻骨,哪得梅花扑鼻香”。惟有这样,才能在破解难题大题

的过程中打好厚实功底,才能取得持续的进步。

(3) 处理好“前沿性”与“基础性”、“直线培训”和“散点培训”、“循序渐进”与“跳跃式”、“培训活动”和“网上自学”的关系。教师授课时,一般讲的是基础性知识,内容的安排通常采用循序渐进的直线方式。但结果往往会出现与竞赛要求“不合拍”的情况。这种情况的出现是正常的,因为除了命题者外,谁也说不清楚未来竞赛中究竟什么知识是必要的。教师的主要作用是为学生的长远发展打下一个扎实的基础。但是,从另一个角度看,如果仅仅恪守这种按部就班的培训程序,不谋求跳跃式学习,很可能离程序设计的前沿知识、离全国和国际信息学奥林匹克活动的前沿越来越远。因此在进行基础课程学习的同时,必须有追逐前沿的选择性学习。

首先要敢于跳跃。有时候心理的障碍比科学上的障碍更难跨越,“敢不敢”的问题比“能不能”的问题更突出。其实在学习中或多或少地都有必要的跳跃,不少人还能够实现比较大的飞跃。例如,我们的许多选手起初并不了解什么是图论、什么是组合数学,但信息学竞赛涉及到了这些较难的相关知识,他们就带着问题学,不仅学懂了,而且还能够应用自如。

其次到网上去寻求帮助。现在,各种各样的试题和算法知识在网上浩如烟海,使得学生学习的大部分时间和空间由课堂转移到互联网上,自主性显著提高,学习方式发生了根本性的变化。对于教师来说,指导要有选择性,不必要也不可能把所有重要的东西都选择好了交给学生,而是应该将直线培训与散点培训相结合,选择部分重要的东西交给学生,让他们自己去探索若干知识点之间的联系,补充自己认为需要补充的知识;对于学生来说,应该对网上的信息有一个科学的价值判断:“这道题是不是有利于夯实基础,有没有需要学习的新知识”,或者“这道题是不是引出带规律性的东西,能不能派生出更多类似的问题”……真正从提高自身的认知水平和编程能力的高度来决定取舍。网上学习的效率取决于如何在浩瀚的信息海洋中选择最有价值的知识,能否构建个性化(符合自己能力结构和兴趣结构)和竞争需要的知识结构。

2. 以问题为出发点、以构造网络式思维模式为基础、以“双轨复式”和“合作学习”为主要方式组织培训活动

(1) 用问题驱动培训活动。学习一定要明确知识的问题背景,学习程序设计知识,尤其如此。编程解决实际问题的过程如下。

第一步:界定问题。即明确试题要求我们做什么,解决什么问题,输出什么结果。

第二步:明确出发点。即明确试题的初始状态是什么,给出了哪些求解条件,其中哪些是显性的,哪些是隐性的,必须无一漏失。

第三步:寻找怎样做的创意。即要求学生独立思考,尽可能多地收集相关资料,尝试各种组合,调动所有的灵感和思考方式。

第四步：集中测试。每人设计三组测试数据，分别从一般情况、边界情况、扩大问题规模 3 个角度测试程序的正确性、时间效率和空间效率。然后交流测试数据，学生间进行互测。这样做既能测出自己程序的正确程度，又能在改善程序的时空效率上取长补短。

其中，第一步和第二步是明确问题的背景，第三步和第四步是解决问题的过程。我们带着第一步和第二步形成的问题寻求相关的知识，并且经过第三步和第四步的实践过程加深对知识的理解，这种问题驱动学习的效果是“本本学习”远不可比的。要拓展知识、提高应用水平也可采取问题驱动的方式。例如，做了一题以后，想一想当前的解法有没有缺陷，有没有更好的算法，它与哪些问题有联系，与哪些知识相关联，还可以拓延出一些更深层次的问题，要解决这些问题还需要哪些知识。

我们的培训首先要培养学生的问题意识和解决问题的能力。创造始于问题，“有了问题才会思考，有了思考才有解决问题的方法，才有找到独立思路的可能”（陶行知）。虽然有问题不一定有创造，但没有问题一定没有创造。

(2) 培训活动应该以构造网络式思维模式为基础。我们应该通过归纳概括和触类旁通、联想外推这两种途径构造知识信息网络。

归纳概括：通过对相关问题（即形式不同、本质类似的一类问题）的归纳，揭示内在的联系，概括出解决类似问题的一般规律，并得到高度抽象、概括的模型，为以后分析问题、设计算法奠定基础，图 1-1 为归纳概括示意图。

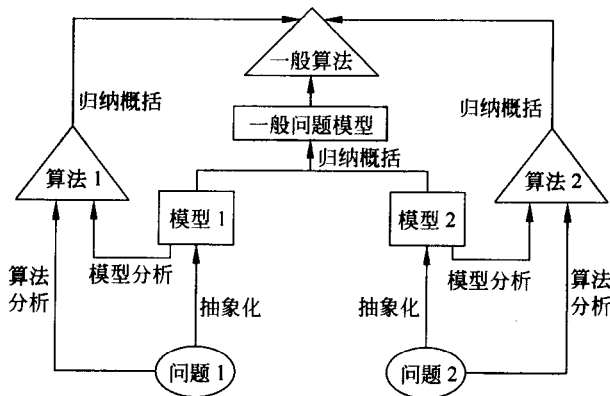


图 1-1

一般来讲，归纳概括出的模型抽象的程度愈高，应用的范围愈广。例如，动态程序设计方法中状态转移方程的一般模式为

$$f_k(s_k) = \underset{u_k \in D_k(s_k)}{\text{opt}} g(f_{k+1}(T_k(s_k, u_k)), u_k)$$

$f_n(s_n) = \text{某个初始值}$ (边界条件)

其中, s_k 是第 k 段的某个状态, u_k 是从 s_k 出发的允许决策集合 $D_k(s_k)$ 中的一个决策, $T_k(s_k, u_k)$ 是由 s_k 和 u_k 所导出的第 $k+1$ 段的某个状态 s_{k+1} , $g(x, u_k)$ 是定义在数值 x 和决策 u_k 上的一个函数, 而函数 opt 表示最优化, 根据具体问题分为 $\max$ 或 $\min$ 。最优指标函数 $f_k(s_k)$ 表示从第 k 段状态 s_k 采用最优策略到过程终止时的最佳效益值。

这个数学模型可谓抽象, 但可用这一模型求解的问题又何止千万。

触类旁通、联想外推: 通过举一反三、由此及彼、触类旁通, 从一个问题拓展出许多新问题。在解决这些新问题的过程中, 进一步锻炼思维, 并通过联想的线索将新问题、新模型、新算法并入知识网。图 1-2 所示为触类旁通、联想外推示意图。

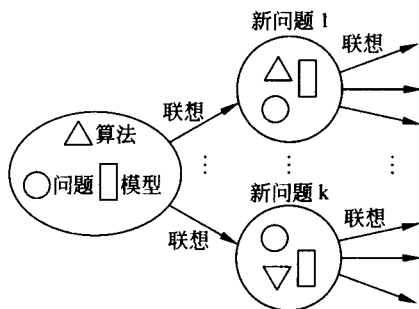


图 1-2

例如, 使用动态程序设计方法求解的一般模式是使用多重循环枚举所有阶段、当前阶段的所有状态和当前状态的所有决策, 并在循环体内通过状态转移方程计算最优解。但是, 学生们在解题实践中不断拓宽了它的思想内涵和应用范围。例如, 对于一些非线性结构的问题, 可以直接根据状态转移方程 $f_k(s_k) = \underset{u_k \in D_k(s_k)}{\text{opt}} g(f_{k+1}(T_k(s_k, u_k)))$, u_k 的递归定义, 自上而下地求解; 如果取消状态转移方程中的最优性要求 (即取消转移方程中的 opt), 则可以计算具有重复子结构的所有路径问题; 对于阶段性和重复子结构特征明显, 但不具备最优子结构特征的最优化问题, 则可以变状态转移方程 $f_k(s_k)$ 为布尔表达式, 将最优化问题转化为判定性问题, 然后递推最优解; 如果是计算图添边后的最短路径, 则可以使用双重动态程序设计; 如果求多条最佳路径, 则可以使用多进程动态程序设计等。

我们在解题过程中要学会对问题适当“外延”, 即通过修改试题条件将问题外延, 主动考虑相近、相关、相对的问题, 自觉地去实践、探索创新。这与一般意义上的题海战术不同。题海战术是一味追求模仿、追求数量、用死记硬背的机械模仿取代从实际出发的逻辑思维和直觉思维, 这样做只会助长学生思维的惰性, 事倍功半。而问题适当外延, 既可加深学生对其内涵的理解, 又可帮助学生通过事物间的联系拓宽知识面。大量的实践

表明,学生的思维能力与解题过程中对各种信息的概括能力以及解题后的信息保持能力成正比。例如,一个推理过程:

$$A \rightarrow B \rightarrow C \rightarrow \cdots \rightarrow G \rightarrow H$$

习惯死记硬背的学生在从“A”到“H”的推理过程中要走过相当“漫长”的路程,解题后的概括难以胜任,只能得到一些孤立的现象,并试图用同样的努力去记住所有的、非本质的东西;而主动外延问题的学生,解题中善于紧缩思维,他们可直接由“A”看到“H”,仅在分析一种现象的基础上,归纳到一串相关相似的现象,把每个题目认作一类题目的代表,并按特定的方式解题。他们自觉地探索各类问题中隐藏着的普遍规律,对题型的逻辑模式和推理概要都能留下深刻印象,且能长久保持。

(3)“双轨复式”和“合作学习”是培训的主要方式。这里讲的“双轨”是指培训沿着两条线:一条是按照数据结构、算法和编程技术的知识结构安排培训计划;一条是为每一类知识配备相关的典型试题和经典解法,既尊重知识结构本身的科学性,又注重创设问题背景,理论联系实际。“复式”是指某些特别重要的内容或较复杂的算法,不能指望一两两次活动就能让学生做到融会贯通、熟练掌握,必须依据量力而行的原则,根据其学习进程和智力发展水平把同一内容分成深度、广度不同的若干片断,一步步地螺旋式上升,最后达到较好掌握。问题过难过易都不能出现在一段较长的时间里,过难会丧失信心,过易则难以激发热情。因此,要把握好节奏,一个阶段一个中心,使学习目标始终与自己“最近发展区”的实际可接受性相适应,构造一个有利于提高学习效率的知识链。

这里应该特别强调的是,培训应该采取合作学习的方式。过去,同学们在培训活动中,通常是你做你的,我干我的,很少交流。我们提倡所有学员的解题方法一律公开,通过研讨算法、集中编程、互测数据等互相合作的方式共同完成学习任务。之所以要这样做,是因为合作能够提高学习的整体效果。信息学奥林匹克竞赛的培训活动属于高层次的学习活动,试题的难度一般比较大,考虑的因素可能比较多。集思广益才容易得出好的算法,群体设计的测试数据相对比较全面,每个个体只有在群体活动中才能比较客观地反映出自己的能力情况,只有在付出与给予中才能提高自己的合作精神和编程能力。历届竞赛活动的成功者往往是那些相容性好、乐于帮助他人并且善于取他人之长的学生。

每个参与培训活动的同学应该调整好自己的心态。如果大家编程解题的水平相当,固然是好,即使有差异,也可以互补,英语好的同学可以帮助翻译网上试题,数学好的同学可以帮助参谋算法,“三人行,则必有我师焉”,许多大题难题的成功解法往往是集体智慧的结果。即便自己“技高一筹”也没有什么了不起,在知识海洋中只不过是“沧海一粟”。况且是不是真行,还得由群体活动的实践来检验。从成本效益的角度看,固守错误概念比一无所知更可怕,因为固守错误概念是个人付出相当多的时间和精力后的结果,从客观角度看,与未付出任何代价的“一无所知”几乎是等价的,因为在竞赛中,错误答案