信息学奥林匹克竞赛指导丛书

吴文虎 主编

信息学奥林匹克竞赛指导

—— 2001竞赛试题解析

吴文虎 王建德 编著

✓ 训练思维能力

✓ 提高解题技巧

4

清华大学出版社

信息学奥林匹克竞赛指导丛书
吴文虎　主编

信息学奥林匹克竞赛指导

——2001竞赛试题解析

吴文虎　王建德　编著

清 华 大 学 出 版 社
北　京

内 容 简 介

本书收集了2001年信息学奥林匹克国际赛、全国赛、组队赛共26道试题，书中对试题进行了类型归纳，并对每一种类型的解题思路作了简要的介绍，对每一道典型试题进行了算法解析，同时给出了详细的解题过程描述，所有解法的正确性和时空效率都通过了竞赛测试数据的验证。另外，还刊载了2001年国际信息学奥林匹克竞赛中国集训队的辅导讲义。书中并未提供直接上机运行的源代码，而是采用比较贴近自然语言的PASCAL语言来描述算法的基本思想和步骤的，这就为读者上机实践留下了空间。

版权所有，翻印必究。
本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

信息学奥林匹克竞赛指导——2001竞赛试题解析/吴文虎主编；吴文虎，王建德编著. —北京：清华大学出版社，2003
(信息学奥林匹克竞赛指导丛书)
ISBN 7-302-06717-1

Ⅰ. 信… Ⅱ. ①吴… ②王… Ⅲ. 计算机课—中小学—解题 Ⅳ G634.673

中国版本图书馆CIP数据核字(2003)第045589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：宋方
印 刷 者：北京牛山世兴印刷厂
装 订 者：北京密云京文制本装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：12.5　**字数**：286千字
版　　次：2003年10月第1版　2003年10月第1次印刷
书　　号：ISBN 7-302-06717-1/TP・5009
印　　数：1～5000
定　　价：19.80元

前　言

信息科学与技术正在对人类社会的发展产生难以估量的深远影响，作为人类总体智慧的结晶，电脑已经成为一种新的现代文化。“计算机的普及要从娃娃做起”已经深入人心，成为“科教兴国”的一项重要内容。

一个国家、一个民族要想不落伍，要想跻身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，说到底也是人才的竞争。电子计算机是现代科学与技术的基础和核心，它的飞速发展，把社会生产力水平提到前所未有的高度，人类进入了信息时代。电脑对人类社会的发展所起的巨大作用，特别是对人类智能的发展所起的促进作用，已为人们普遍认识到。计算跟语言一样是人类社会每时每刻都不可缺少的。现在，人类已经拥有了帮助自己进行复杂计算与思维的工具，电子计算机起到了人脑延伸的作用。以往历史上的技术革命，只能起到创造和改进工具，用机器代替人的体力的作用；而计算机则是把人从重复性的或有固定程式的脑力劳动中解放出来，使自己的智能获得空前的发展。作为“人类通用智力工具”的计算机在开发人类智能方面所起的无与伦比的作用不容忽视。这也就是计算机与基础教育相结合，能够成为当今世界的大趋势的一个原因。从信息社会要求人才具备的科学素养看，数学、物理学、化学、生命科学和信息科学是五大支柱，这正是联合国教科文组织倡导举行五项国际学科奥林匹克竞赛的内容。

国际信息学奥林匹克（International Olympiad in Informatics，简称 IOI）始于 1989 年，到 2002 年已成功地举办了 14 届。这是一种智力与应用计算机能力的大赛。从益智的角度看，是用电脑帮助开发人脑，重在提高思维能力，培养创新意识。在中国队的训练中强调德智体美全面发展；心态上自立、自尊、自信、自强，要怀着中华民族的自豪感和自信心去参赛；这种心态是学习、训练和取胜的重要条件。14 届比赛，中国队每届都取得了名列前茅的好成绩，55 人次参赛，夺得 55 块奖牌，其中金牌 29 块、银牌 15 块、铜牌 11 块。特别是 IOI’95（荷兰）突破了前 6 届比赛女孩与金奖无缘的纪录，两名中国女选手荣登金牌领奖台。在 IOI’96（匈牙利）上中国队又实现了全金的突破，四名选手，每人获得一块金牌。

从大局看，竞赛不是目的，是推动普及的手段，我们的目的只有一个：科教兴国。竞赛活动带有因材施教、因材施测的特点。普及是有层次的，与学科竞赛有关的普及活动，对青少年而言属于比较高的层次，当然就有相当的难度。我们编写的这套竞赛指导丛书，涉及程序设计语言、常用算法、组合数学、图论、人工智能搜索等的基本知识和基本方法。这些理论知识往往都是通过竞赛当中的一些实例来讲解的，重点放在解题思路上，书中有许多题目比较新颖，很难去套固定算法或固定模式，这中间有些招数

是选手们想出来的。从中可以看出信息学奥林匹克要求创新，鼓励创新。当然，书中给出的解法，对青少年读者而言，我们希望仅仅起到抛砖引玉的作用，并且热切盼望引出更多的“玉”来。作为老师，我和王建德都这样想，“精心育桃李，热望青胜蓝”就是我们编写这套丛书的初衷。

国际信息学奥林匹克中国队总教练
清华大学计算机系教授博士生导师
吴文虎
2003年6月

试读结束：需要全本请在线购买： www.ertongbook.com

目　　录

第1章 数学分析

在去年竞赛中，数学分析类试题频繁出现，由此可以看出数学和程序设计之间的“孪生关系”：数学中难以用笔和纸推算的问题需要借助计算机解决；而编程者要解决此类问题，需要有坚实的数学功底和灵活的应变能力，能够对运算对象进行组合分析——怎样计算具有某种特性的对象个数，怎样枚举这些对象。信息学竞赛中离散数和有限数一类试题激增，正说明了信息学与数学的依赖关系日益凸现，数学需要反映计算机进行计算、检索、记忆、决策的原理和机制，信息学的发展需要现代数学的支撑。两门学科的整合是国际中学理科教育发展的大趋势。

1.1 解 方 程

使用计算机解方程，与其说是考核选手的编程技术，不如说是考核选手的数学技巧和能力。解题的关键是通过数学分析得出计算公式和公式中变量的取值范围，在此基础上通过顺序查找或分治法枚举变量的可能值，将符合条件的变量代入表达式，即可得出问题的解。这是编程解数学题的一般思路，也是程序设计竞赛与数学竞赛的区别所在。

例题1 反正切函数的应用(全国赛)

反正切函数可展开成无穷级数，有如下公式：

$$\arctan(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{2n+1} \qquad (其中 0 \leqslant x \leqslant 1) \qquad 公式(1)$$

使用反正切函数计算 π 是一种常用的方法。例如，最简单的计算 π 的方法

$$\begin{aligned}\pi &= 4\arctan(1) \\ &= 4\left(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\frac{1}{9}-\frac{1}{11}+\cdots\right)\end{aligned} \qquad 公式(2)$$

然而，这种方法的效率很低，但我们可以根据角度和的正切函数公式

$$\tan(\alpha+\beta) = \frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)} \qquad 公式(3)$$

通过简单的变换得到

$$\arctan(p)+\arctan(q) = \arctan\left(\frac{p+q}{1-pq}\right) \qquad 公式(4)$$

利用这个公式，令 $p=\frac{1}{2}, q=\frac{1}{3}$，则 $\frac{p+q}{1-pq}=1$，有

$$\arctan\left(\frac{1}{2}\right)+\arctan\left(\frac{1}{3}\right) = \arctan\left(\frac{\frac{1}{2}+\frac{1}{3}}{1-\frac{1}{2}\cdot\frac{1}{3}}\right) = \arctan(1)$$

使用$\frac{1}{2}$和$\frac{1}{3}$的反正切来计算 arctan(1)，速度就快多了。

我们将公式(4)写成如下形式

$$\arctan\left(\frac{1}{a}\right)=\arctan\left(\frac{1}{b}\right)+\arctan\left(\frac{1}{c}\right)$$

其中 a、b 和 c 均为正整数。

问题是，对于每一个给定的 $a(1\leqslant a\leqslant 60\,000)$，求 $b+c$ 的值。要保证对于任意的 a 都存在整数解，如果有多个解，则要求给出 $b+c$ 最小的解。

输入与输出

输入文件(arctan. in)中只有一个正整数 a，其中 $1\leqslant a\leqslant 60\,000$。

输出文件(arctan. out)中只有一个整数，为 $b+c$ 的值。

输入输出样例

输入

1

输出

5

题解

这是一道有趣的三角函数题。已知 a 的值，根据公式

$$\arctan\left(\frac{1}{a}\right)=\arctan\left(\frac{1}{b}\right)+\arctan\left(\frac{1}{c}\right)=\arctan\left(\frac{\frac{1}{b}+\frac{1}{c}}{1-\frac{1}{bc}}\right)=\arctan\left(\frac{b+c}{bc-1}\right)$$

计算 $b+c$ 的最小值。由公式可以看出

$$\frac{1}{a}=\frac{b+c}{bc-1}$$

即 $bc-1=a(b+c)$，对该式进行变换，得出

$$(c-a)(b-a)=a^2+1 \qquad \text{公式(1)}$$

即
$$b+c=\frac{a^2+1}{b-a}+a+b=2a+b-a+\frac{a^2+1}{b-a}$$

在上式中，$b-a$ 是一个变化的数值。为了取得 $b-a$ 的数值范围，不妨设 $b<c$。由 $(b-a)^2<(b-a)(c-a)=a^2+1$ 可以得出，$1\leqslant b-a\leqslant a$。对于 $b+c=2a+b-a+\frac{a^2+1}{b-a}$来说，$b-a$ 愈大，$b+c$ 愈小。将可被 a^2+1 整除的最大的 $b-a$ 代入 $2a+b-a+\frac{a^2+1}{b-a}$中，即可得出 $b+c$ 的最小值。由此得算法：

```
p←a²+1;
for k←a down to 1 do          {按照递减顺序枚举 b—a 的所有可能值(k=b—a)}
  if p mod k=0                {若 b—a 可被 a²+1 整除，则输出 b+c 的最小值}
```

then begin 输出 $2a+k+\frac{a^2+1}{k}$;break;end;{then}

例题 2　一元三次方程求解(分区联赛)

有形如 $ax^3+bx^2+cx+d=0$ 这样的一个一元三次方程。给出该方程中各项的系数(a,b,c,d 均为实数),并约定该方程存在三个不同实根(根的范围在-100～100 之间),且根与根之差的绝对值为大于等于 1。

要求由小到大依次在同一行输出这三个实根(根与根之间留有空格),并精确到小数点后 2 位。

提示:设方程 $f(x)=0$,若存在两个数 x_1 和 x_2,且 $x_1<x_2$,$f(x_1)\cdot f(x_2)<0$,则在(x_1,x_2)之间一定有一个根。

输入输出样例

输入

1　-5　-4　20

输出

-2.00　2.00　5.00

题解

这是一道有趣的解方程题。为了便于求解,将原方程

$$f(x)=ax^3+bx^2+cx+d=0$$

变换成

$$f'(x)=x^3+b'x^2+c'x+d'=0$$

的形式$\left(其中\ b'=\left\lfloor\frac{b}{a}\right\rfloor, c'=\left\lfloor\frac{c}{a}\right\rfloor, d'=\left\lfloor\frac{d}{a}\right\rfloor\right)$,$f(x)$ 和 $f'(x)$的根不变。设 x 的值域(-100～100 之间)中有 x,其左右两边相距 0.0005 的地方有 x_1和 x_2两个数,即

$$x_1=x-0.0005 \qquad x_2=x+0.0005$$

x_1和 x_2间的距离(0.001)满足精度要求(精确到小数点后 2 位)。若出现图 1.1 所示的两种情况之一:

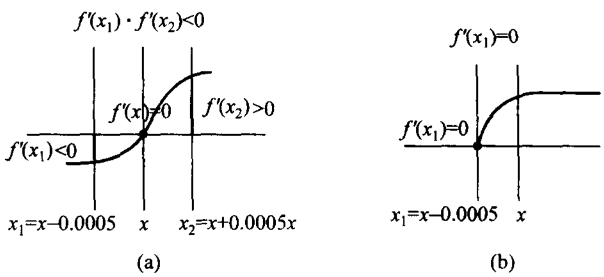

图　1.1

则确定 x 为 $f'(x)=0$ 的根。有两种方法来计算 $f'(x)=0$ 的根 x:

1. 枚举法

根据根的值域和根与根之间的间距要求(大于 1),我们不妨将根的值域扩大 100 倍($-10\,000 \leqslant x \leqslant 10\,000$),依次枚举该区间的每一个整数值 x,并在题目要求的精度内设定区间:$x_1=\dfrac{x-0.05}{100}$,$x_2=\dfrac{x+0.05}{100}$。若区间端点的函数值 $f'(x_1)$和 $f'(x_2)$异号或者在区间端点 x_1 的函数值 $f'(x_1)=0$,则确定$\dfrac{x}{100}$为 $f'(x)=0$ 的一个根。由此得出算法:

```
输入方程中各项的系数 a,b,c,d;
b←b/a; c←c/a; d←d/a; a←1;              {将方程变换为 x³+b'x²+c'x+d'=0 的形式}
for x←-10000 to 10000 do                {枚举当前根乘以 100 的可能范围}
  begin
    x1←(x-0.05)/100;    x2←(x+0.05)/100; {在题目要求的精度内设定区间}
    if (f(x1) * f(x2)<0) or (f(x1)=0)   {若在区间两端的函数值异号或在 x1 处的函
                                          数值为 0,则确定 x/100 为根}
    then write(x/100: 0: 2,' ');
end;{for}
```

其中函数 $f(x)$用来计算 x^3+bx^2+cx+d 的值:

```
function f(x: extended): extended;      {计算 x³+bx²+cx+d}
begin
  f←x * x * x+b * x * x+c * x+d;
end;{f}
```

2. 分治法

用于枚举根的值域中的每一个整数 $x(-100\leqslant x\leqslant 100)$,由于根与根之差的绝对值大于等于 1,因此设定搜索区间$[x_1,x_2]$,其中 $x_1=x$,$x_2=x+1$。若

(1) $f'(x_1)=0$,则确定 x_1为 $f'(x)$的根;

(2) $f'(x_1)\cdot f'(x_2)>0$,则确定根 x 不在区间$[x_1,x_2]$内,设定$[x_2,x_2+1]$为下一个搜索区间;

(3) $f'(x_1)\cdot f'(x_2)<0$,则确定根 x 在区间$[x_1,x_2]$内。

问题是,如果确定根 x 在区间$[x_1,x_2]$内的话($f'(x_1)\cdot f'(x_2)<0$),那么如何在该区间内找到根的确切位置。采用二分法,将区间$[x_1,x_2]$分成左右两个子区间:左子区间$[x_1,x]$和右子区间$[x,x_2]$$\left(\text{其中 } x=\dfrac{x_1+x_2}{2}\right)$。

如果 $f'(x_1)\cdot f'(x)\leqslant 0$,则确定根在左区间$[x_1,x]$内,将 x 设为该区间的右指针($x_2=x$),继续对左区间进行对分。

如果 $f'(x_1)\cdot f'(x)>0$,则确定根在右区间$[x,x_2]$内,将 x 设为该区间的左指针($x_1=x$),继续对右区间进行对分。

试读结束:需要全本请在线购买: www.ertongbook.com

上述对分过程一直进行到区间的间距满足精度要求为止($x_2-x_1<0.001$)。此时确定 x_1 为 $f'(x)$的根。由此得出算法：

```
输入方程中各项的系数 a,b,c,d;
b←b/a; c←c/a; d←d/a;a←1;         {将方程变换为 x³+b'x²+c'x+d'=0 的形式}
for x←-100 to 100 do                {枚举每一个可能的根}
begin
  x1←x;x2←x+1;                      {确定根的可能区间}
  if f(x1)=0                        {若 x1 为根,则输出}
  then write(x1: 0: 2,' ')
  else if (f(x1) * f(x2)<0)         {若根在区间[x1,x2]中}
      then begin
           while x2-x1>=0.001 do    {若区间[x1,x2]不满足精度要求,则循环}
           begin
             xx←(x2+x1)/2;          {计算区间[x1,x2]的中间位置}
             if f(x1) * f(xx)<=0    {若根在左区间,则调整右指针}
             then x2←xx
               else x1←xx;          {若根在右区间,则调整左指针}
           end;{while}
           write(x1: 0: 2,' ');     {区间[x1,x2]满足精度要求,确定 x1 为根}
      end;{then}
end;{for}
```

其中 $f(x)$的函数说明如枚举法所示。

1.2 逻辑推理

将运算对象定义为逻辑变量,按照题意要求和对象间的关系进行布尔运算,求出问题的解。这里提醒大家的是要注意理解题意。解题过程是一个不断发掘已知条件向目标靠拢的过程,因此不能视题目的文字描述为鸡肋,不屑一顾。有时候文字描述中蕴藏了丰富的信息,对解题起到了决定性的作用。只有在全面正确地理解题意的基础上,才能准确定义对象间的关系,并通过布尔运算进行合乎逻辑的推理。

例题 3 聪明的学生(组队赛)

一位教授逻辑学的教授有三名非常善于推理且精于心算的学生 A、B 和 C。有一天,教授给他们三人出了一道题：教授在每个人脑门上贴了一张纸条并告诉他们,每个人的纸条上都写了一个正整数,且某两个数的和等于第三个。于是,每个学生都能看见贴在另外两个同学头上的整数,但却看不见自己的数。

这时,教授先对学生 A 发问了:“你能猜出自己的数吗?”A 回答:“不能。”

教授又转身问学生 B:“你能猜出自己的数吗?”B 想了想,也回答:“不能。”

教授再问学生 C 同样的问题,C 思考了片刻后,摇了摇头:“不能。”

接着,教授又重新问 A 同样的问题,再问 B 和 C,……,经过若干轮的提问之后,当教授再次询问某人时,此人突然露出了得意的笑容,把贴在自己头上的那个数准确无误地报了出来。

现在,如果告诉你:教授在第 N 次提问时,轮到回答问题的那个人猜出了贴在自己头上的数是 M,你能推断出另外两个学生的头上贴的是什么数吗?

提示:在没有人猜出自己头上的数之前,大家对教授提问的回答始终都是“不能”;而且除此之外,在 A、B、C 之间是没有进行任何信息交流的。也就是说,每个人推断的依据仅仅是另外两个人的头上数,以及大家对教授的提问所做出的否定回答。

教授总是从学生 A 开始提问的。

你可以假定,这三个足够聪明的学生能够根据已知的条件在最早的轮次猜出自己的数,并且永远都不会猜错。

稍经分析和推理,你将得出以下结论:总是头上贴着最大的那个数的人最先猜出自己头上的数。

输入与输出

输入文件为 guess. in。该文件包括若干组测试数据,其中的每一行代表一组测试数据,由两个整数 n 和 m 组成(即在教授第 N 次提问时,轮到回答问题的那个人猜出了贴在自己头上的数是 m)。两个数之间用空格分隔开。最后,由“−1　−1”组成的一行标志着输入数据的结束。

$0<n<500$;　$0<m<30\,000$

输出文件为 guess. out。按照输入文件中的顺序依次给出各组数据的结果。

文件中对应每组数据输出的第一行是一个整数 p,是可能情况的总数。接下来的 p 行,每一行包括三个数,分别为贴在 A、B、C 头上的三个数。输出时,所有解按照 A 头上的数增序排列;在 A 头上的数相同的情况下,按照 B 头上的数增序排列。

输入输出样例

guess. in	
5	8
3	2
2	3
−1	−1

guess. in		
3		
2	8	6
5	8	3
6	8	2
1		
1	1	2
1		
2	3	1

题解

这是一道逻辑推理题。在三个学生头上贴三个正整数,他们可以根据题目给出推理的规则依次循环猜测自己头上的数,直到有一人猜出。题目给出了最大数 M 及猜测轮次

N,要求按照从小到大的顺序输出三个学生头上的数的所有可能情况。

本题的解题线索隐蔽在这些看似平淡的文字中。而问题的关键在于弄清什么情况下可以猜出自己头上的数。因此,我们要围绕这个目的仔细挖掘题目的条件。

试题明确提出“某两个数的和等于第三个”。也就是说,每个学生头上的数不是其余两数之和,就是其余两数之差,两者必居其一。同时,试题告知“总是由头上贴着最大的那个数的人最先猜出自己头上的数”,即最先猜中的数是其余两数之和。

试题告知,在某个同学头上的数被猜出之前,每个学生回答均是“不能”。从表面上看,这种一致否定的回答似乎未提供足够的猜测依据,其实不然。若是将它与猜测次数联系起来,则其中蕴涵的信息量却相当多。

每一个人能够根据已知条件在最早的轮次猜出自己头上的数,并且永远都不会猜错,即试题给出的猜测次数是最少的。

由此可见,整个推理过程是以答者否定的回答、其余两人头上的数和猜测的最少次数作为依据的。我们根据这些线索模拟答者的心理活动,作出他头上的数是否不为其余两数之差的结论。下面,我们来具体分析这一过程:

设甲、乙、丙三人头上的数分别为 x,y,z。$f(x,y,z,t)$表示能否有人恰好在第 t 次提问时猜出自己头上的数(恰好猜出的定义是一定能在第 t 次猜出又没有人能在此前猜出)。为方便起见,不妨设甲在第 t 次提问时猜出自己头上的数为 x。显然,若 $y=z$,则 $f(x,y,z,1)=\text{true}$(因为 x,y,z 属于正整数,所以 $x=y+z$)。以上仅仅是一种特例,一般的情况则是:若甲恰在第 t 次提问时猜出,则他的猜测过程如下。

若甲猜测 $x=y-z(y>z)$,则根据提示,乙应恰在 $t-2$ 次提问时猜出 $y=x+z$(因为每一个人能够根据已知的条件在最早的轮次猜出自己的数,并且永远都不会猜错),如果乙未猜出,则 $y\neq x+z$,即 $x=y+z$。由此,我们可以得到 $f(x,y,z,t)=f(y-z,y,z,t-2)(t\geqslant 3)$。

若甲猜测 $x=z-y(z>y)$,则根据提示,丙应恰在 $t-1$ 次提问时猜出 $z=x+y$(同上理),如果丙未猜出,则 $z\neq x+y$,即 $x=y+z$。由此,我们可以得到 $f(x,y,z,t)=f(z-y,y,z,t-1)(t\geqslant 2)$。

在函数 f 的迭代过程中,参数是递减的,而根据题意,所有参数都应该为正整数,因此算法的有穷性可以得到保证。

我们依次设 $x=m$、$y=i$、$z=m-i(1\leqslant i\leqslant m-1)$,代入 f 函数,计算出三个学生在 t 次猜出自己的数的所有可能情况。由此得出算法:

```
for  i←1 to m−1 do                    {枚举另一人头上的所有可能解}
  begin
    j←(n−1) mod 3+1;                  {设置猜数人的序号为j,其余两人的序号为k和6−j−k}
    k←2−⌊j/2⌋;
  a[j]←m;  a[k]←i;  a[6−j−k]←m−i;
  t←n; f←false;                       {猜测次数和成功标志初始化}
  repeat
    if t≤0 then break;                {若不能在t次内猜出,则失败退出}
```

```
case j of                          {最大数为 a[j],能否恰在 t 次猜出}
1: begin
     if a[2]=a[3]
       then begin
         if t=1 then f←true;       {猜测成功}
         break;
         end{then}
       else if a[2]>a[3]           {第 t-2 次猜最大数 a[2]=a[1]+a[3]}
           then begin
               j←2;  a[1]←a[2]-a[3];  t←t-2;
               end{then}
           else begin              {第 t-1 次猜最大数 a[3]=a[1]+a[2]}
               j←3;  a[1]←a[3]-a[2];  t←t-1;
               end;{else}
     end;{1}
2: begin
     if a[1]=a[3]
     then begin
         if t=1 then f←true;       {猜测成功}
         break;
         end{then}
     else if a[1]>a[3]             {第 t-1 次猜最大数 a[1]=a[2]+a[3]}
         then begin
             j←1;  a[2]←a[1]-a[3];  t←t-1;
             end{then}
         else begin                {第 t-2 次猜最大数 a[3]=a[2]+a[1]}
             j←3;  a[2]←a[3]-a[1];  t←t-2;
             end;{else}
     end;{2}
3: begin
     if a[1]=a[2]
     then begin
         if t=1 then f←true;       {猜测成功}
         break;
         end{then}
     else if a[1]>a[2]             {第 t-2 次猜最大数 a[1]=a[2]+a[3]}
         then begin
             j←1;  a[3]←a[1]-a[2];  t←t-2;
             end{then}
         else begin                {第 t-1 次猜最大数 a[2]=a[3]+a[1]}
             j←2;  a[3]←a[2]-a[1];  t←t-1;
             end;{else}
```

```
        end;{3}
      end;{case}
    until false;
    if f then                                {若恰在第 t 次猜出,则输出可行解}
      begin
        a[j]←m;   a[k]←i;   a[6−j−k]←m−i;
        输出 a[j],a[k]和 a[6−j−k];
      end;{then}
end; {for}
```

由于函数 f 具有明显的递归定义,上述算法也可以用递归方式来实现。

1.3 初等数论

初等数论是用初等数学方法来研究整数的整除、不定方程、同余式等方面问题的数论分支。其内容包括辗转相除法、因数、倍数、公因数、公倍数、最大公因数、最小公倍数、素数、合数等。初等数论中难以用笔和纸推算的问题可以借助计算机来解决。

例题 4 最大公约数与最小公倍数问题(分区联赛)

输入两个正整数 x_0,y_0($2 \leqslant x_0 < 100\,000$,$2 \leqslant y_0 \leqslant 1\,000\,000$),求出满足下列条件的 p、q 的个数:

条件:(1) p、q 是正整数;

(2) 要求 p、q 以 x_0 为最大公约数,以 y_0 为最小公倍数。

试求:满足条件的所有可能的两个正整数的个数。

输入输出样例

输入 $x_0=3$ $y_0=60$

输出 4

说明:(不用输出)此时的 p、q 值分别为:

```
 3   60
15   12
12   15
60    3
```

所以,满足条件的所有可能的两个正整数的个数共 4 种。

题解

1. 如何判别两个自然数互质

要判别自然数 x 与 y 是否互质,必须先计算 x 和 y 的最大公约数。如果两者的最大公约数为 1,则表明 x 与 y 互质。设 $\gcd(x,y)$ 为 x 和 y 的最大公约数

$$\gcd(x,y)=\begin{cases}x & (y=0)\\ \gcd(y,x \bmod y) & (x>y)\end{cases}$$

通过函数 gcd(x,y)来求最大公约数的代码如下：

```
function gcd(x,y: longint): longint;
  begin
    if y=0 then gcd←x
          else gcd←gcd(y,x mod y);
  end;{ gcd }
```

显然,gcd(x,y)=1 是 x 与 y 互质的标志。除 1 以外的任何自然数 n 都可以惟一地分解成质数的乘积

$n=p_1^{l_1}\cdot p_2^{l_2}\cdot\cdots p_k^{l_k}$ （质数 $p_1<p_2<\cdots<p_k$,质数的次幂 $l_1,l_2,\cdots,l_k>0$）

2. 如何计算以 x_0 为最大公约数,以 y_0 为最小公倍数的(p,q)个数

首先,必须明确几个问题

（1）若 y_0 不能被 x_0 整除,则不存在以 x_0 为最大公约数、以 y_0 为最小公倍数的(p,q)；

（2）若 $x_0=y_0$,则 $p=q=x_0=y_0$,即满足条件的 p 和 q 仅为一对(x_0,y_0)；

（3）若 $x_0\neq y_0$ 且 y_0 能被 x_0 整除,则由于 p 和 q 呈自反关系(如果(p,q)以 x_0 为最大公约数、以 y_0 为最小公倍数的话,则(q,p)亦是以 x_0 为最大公约数、以 y_0 为最小公倍数的),因此满足条件的 p 和 q 一定为偶数对。

那么,如何求(p,q)的个数呢？设

$$\frac{y_0}{x_0}=n=1\cdot p_1^{l_1}\cdot p_2^{l_2}\cdot\cdots p_k^{l_k}$$

显然,$p_0=x_0$ 和 $q_0=x_0\cdot p_1^{l_1}\cdot p_2^{l_2}\cdot\cdots\cdot p_k^{l_k}=y_0$ 满足条件；$p_1=x_0\cdot p_1^{l_1}$ 和 $q_1=x_0\cdot p_2^{l_2}\cdot\cdots\cdot p_k^{l_k}$ 满足条件；$p_2=x_0\cdot p_2^{l_2}$ 和 $q_2=x_0\cdot p_1^{l_1}\cdot p_3^{l_3}\cdot\cdots p_k^{l_k}$ 满足条件；…；$p_k=x_0\cdot p_k^{l_k}$ 和 $q_k=x_0\cdot p_1^{l_1}\cdot p_3^{l_3}\cdot\cdots\cdot p_{k-1}^{l_{k-1}}$ 亦满足条件。由于 $p_1^{l_1}$ 的关系自反((p,q)=(q,p)),因此若$\frac{y_0}{x_0}$分解出 k 个质数的乘积的话,则满足条件的 p 和 q 的个数有$\begin{cases}2k=4 & (k=2)\\ 2(k+1) & (k>2)\end{cases}$

例如,$x_0=4$,$y_0=24$

由于$\frac{y_0}{x_0}=\frac{24}{4}=6=2\times3$,因此满足条件的($p$、$q$)有 4 对：

(4,24)(24,4)　　(8,12)(12,8)　　$8=4\times2$　　$12=4\times3$

又如 $x_0=4$,$y_0=420$

由于$\frac{y_0}{x_0}=105=3\times5\times7$,因此满足条件的($p$、$q$)有 8 对：

(4,420)	(420,4)		
(12,140)	(140,12)	$12=4\times3$	$140=4\times5\times7$
(20,84)	(84,20)	$20=4\times5$	$84=4\times3\times7$
(28,60)	(60,28)	$28=4\times7$	$60=4\times3\times5$

试读结束：需要全本请在线购买： www.ertongbook.com