

国际国内大中学生 程序设计竞赛辅导

王建德 / 编
施伯乐 / 审

• 南 京 师 范 大 学 出 版 社 •

国际国内大中学生 程序设计竞赛辅导

主 审：施伯乐
编 写：王建德



523518

南京师范大学出版社

图书在版编目(CIP)数据

国际国内大中学生程序设计竞赛辅导 / 王建德著 .
—南京: 南京师范大学出版社, 2000.3
ISBN 7-81047-464-2/TP·6

I. 国… II. 王… III. 程序设计—自学参考资料
IV. TP311.1

中国版本图书馆 CIP 数据核字(2000)第 04254 号

南京师范大学出版社出版发行

(江苏省南京市宁海路 122 号 邮编 210097)

全国各地新华书店经销 南京京新印刷厂印刷

*

开本 787×1092 1/16 印张 17 字数 424 千

2000 年 3 月第 1 版 2000 年 3 月第 1 次印刷

印数: 1—8000

定价: 20.00 元

(南京师大版图书若有印、装错误可向承印厂退换)

序 言

目前,国际上面向大中学生的计算机竞赛很多,其中国际奥林匹克信息学竞赛和国际大学生程序设计竞赛两项赛事历时久远,参与国最多,颇具影响力和权威性。

1. 国际奥林匹克信息学竞赛(简称 IOI)

该项竞赛是联合国教科文组织倡导的五大学科竞赛之一,面向全世界的中学生,每年有五十余个国家派队参赛。我国政府十分重视这项活动,将该赛列入中国自然科学基金会资助项目,由中国科协组织一年一届的全国奥林匹克信息学竞赛(简称 NOI),每省市选派三名优秀中学生(包括一名女生)参赛。由全国赛的前十名选手组成中国集训队,经过半年培训,最终由国际奥林匹克信息学竞赛中国组队赛(简称 CIOI)遴选出 4 名中学生组成中国代表队参加当年的 IOI。

2. 国际大学生程序设计竞赛(简称 ACM/ICPC)

该项竞赛由美国计算机协会(ACM)主办,参赛对象为世界各国本科三年级到硕士研究生二年级的学生。竞赛分层次进行,先由国内预选,然后由各国选派代表队参加洲际的竞赛,最后由洲际竞赛选派代表队参加美国的总决赛。这项赛事受到世界各国欢迎,自 1977 年以来已连续举办 23 届。1998 年参加 21 个区域竞赛的代表队已达 1450 支,遍布五大洲。我国每年有十几所大学参与这项活动。

值得一提的是,由于计算机网络的信息传播作用,使得这两项活动的意义远远超越了竞赛本身。各国竞赛、区域竞赛乃至国际竞赛的试题竞相登录 internet 网。在全球各个角落,人们可以通过上网随时查询异国的竞赛资料,只要移动鼠标和键盘就可以置身于任何一场竞赛之中。众多试题在网上交流,不仅为直接参与这两项活动的单位提供了取之不竭、用之不尽的信息资源,而且许多大学在进行软件课程教学时也纷纷从网上下载试题,作为实例或考题。国际奥林匹克信息学竞赛和国际大学生程序设计竞赛对世界计算机教育产生了推波助澜的积极影响。

这两项竞赛活动之所以受到欢迎,不仅是因为它为广大大中学生提供了一个一展身手的机会,更重要的原因是竞赛试题编得相当有水平,对学生颇具吸引力:

知识基础面宽,涵盖了许多科学知识和经典算法。例如:’99 ACM/ICPC 竞赛中的“弹孔”、“森林防护栏”、“机器人”等题,涉及立体或平面的几何计算;’99 CIOI 竞赛中的“01 统计”,涉及组合数学的计数知识;“家园”、“补丁 VS 错误”、“拯救大兵瑞恩”等题,涉及图论中的网络流运算和隐式图的搜索;’99 NOI 竞赛中的“棋盘分割”,涉及动态程序设计的方法,等等。

所有试题源于现实生活中的问题,既有趣又新颖,要求解题者面对实际,能寻求一种合适的解决方法,而不在乎这种方法本身的难易程度如何。虽然解题需要一定的基础知识,但不能死套条条框框和现成模式。它要求解题者有思想、有方法,有基础训练的经历和经验,能洞察出其间隐藏着的规律,灵活机动地解题。

本书收集了1999年ACM/ICPC,NOI,CIOI,IOI四大竞赛的26门试题及其解析。本书的程序反反复复编写了多次,并通过严格的数据测试,具有较高的时间效率和空间效率。算法分析也写得不错,通篇围绕解题这个中心,深入浅出,循循善诱,力求通俗简练,反映出作者在计算机教育和编程技术方面的扎实功底。

有人说,编程解题是应用计算机的艺术。既然是艺术,就必须见多识广,善于揣摩别人的思想方法,多实践、多体会。编程艺术体现在编程过程之中,而编程过程可以归结于两项主要活动,即人们对所要解决的问题及其相关事物的认识和基于这种认识所进行的描述。

所谓认识是指:在试题所要处理的问题域范围内,通过人的思维对问题域客观存在的事物以及所要解决的问题产生正确的认识和理解,包括弄清事物属性、行为及其彼此之间的关系,并在归纳、联想、类比的创造性过程中构造解题方法。

所谓描述是指:使用适当的数据结构和编程语言,把人们对问题域中事物的认识、对问题及其方法的认识描述出来,最终产生一个计算机能够理解和执行的系统实现。

认识和描述的正确性由测试环节加以验证。即通过设计测试数据(特别是边界值)和测试方法尽量找出认识和描述过程中所有可能出错的地方,修正和完善程序。

应当特别指出的是,在上述过程中尤其要注意培养自己的观察力和想像力。著名科学家爱因斯坦曾经说过:“想象力比知识更重要,因为知识是有限的,而想像力概括着世界上的一切,推动着进步,并且是知识的源泉。”由此可见,作为大学生不仅要有丰富的书本知识,而且还要有运用知识于不同问题情景的实践能力和社会适应力,更要有独立思考、提出问题、拓延思路、探索新知识的创造力。这些知识和能力是你在未来知识经济时代中最能体现“自我”的个人价值,一旦与社会需要融合,将形成一种任何生产资料都无法取代、能产生巨大效益的智力资源。

最后,让我介绍一下这本书的作者:王建德——计算机特级教师。90年代以来,他先后培养的9名中学生在国际奥林匹克信息学竞赛中获六金、一银、二铜;他执笔编写了《实用算法与程序设计》、《人工智能搜索与程序设计》、《图论与程序设计》、《组合数学与程序设计》等九本专著,分别由北京大学出版社、清华大学出版社、复旦大学出版社、电子工业出版社出版。“实践出真知。”在本书的最后一章,作者对自己长期从事培训活动的经历作了一次理性的思考,提出了“培训的目标是让学生学会怎样学习”的观点,谈了自己分阶段教学(基础性课程阶段、发展性课程阶段和研究性课程阶段)的一些作法,诉体验于笔端,通篇文章取材翔实,

立意新颖,论据充分,对参与培训活动的广大师生颇有参考价值。

最后,祝作者在今后的学习和工作中取得更大成绩!

中国计算机学会常务理事、上海市计算机学会理事长

教育部计算机教学指导委员会副主任

复旦大学首席教授、博士生导师

施伯乐

1999年11月

目 录

第一章 '99 国际大学生计算机程序设计竞赛试题解析

§ 1.1 蜜蜂的繁殖	(1)
§ 1.2 弹孔	(7)
§ 1.3 骰子问题	(16)
§ 1.4 森林防护栏	(26)
§ 1.5 在佛维吉斯坦的贸易	(34)
§ 1.6 机器人	(41)
§ 1.7 电子邮件的传送	(51)
§ 1.8 积水问题	(65)

第二章 '99 国际奥林匹克信息学竞赛中国组队赛试题解析

§ 2.1 01 统计	(70)
§ 2.2 补丁 VS 错误	(79)
§ 2.3 家园	(87)
§ 2.4 数字游戏	(98)
§ 2.5 月亮之眼	(110)
§ 2.6 拯救大兵瑞恩	(116)

第三章 '99 全国青少年信息学(计算机)奥林匹克竞赛试题解析

§ 3.1 01 串	(126)
§ 3.2 钉子和小球	(131)
§ 3.3 生日蛋糕	(136)
§ 3.4 棋盘分割	(141)
§ 3.5 最优连通子集	(148)
§ 3.6 内存分配	(153)

第四章 '99 国际奥林匹克信息学竞赛试题解析

§ 4.1 花店柜台布置	(165)
§ 4.2 隐藏着的关键字	(169)
§ 4.3 地下城市	(178)
§ 4.4 交通灯	(192)
§ 4.5 平分薄片	(205)
§ 4.6 平坦的大地	(213)

第五章 在培训活动中引导学生“学会学习”

§ 5.1 学会学习的背景——新技术革命的呼唤和教育改革的需要	(225)
§ 5.2 学会学习的动因——激发乐学精神	(227)
§ 5.3 学会学习的示范——教师循循善诱	(231)
§ 5.4 学会学习的目标——学生成为创造的实践主体	(235)
§ 5.5 附录	(242)

(1)	1.1.1
(7)	1.1.2
(10)	1.1.3
(20)	1.1.4
(34)	1.1.5
(41)	1.1.6
(51)	1.1.7
(53)	1.1.8

附录 2 1999 年全国奥林匹克竞赛题选编

(70)	1.1.1
(72)	1.1.2
(87)	1.1.3
(98)	1.1.4
(110)	1.1.5
(116)	1.1.6

附录 3 1999 年全国青少年奥林匹克竞赛题选编

(130)	1.1.1
(131)	1.1.2
(136)	1.1.3
(141)	1.1.4
(148)	1.1.5
(153)	1.1.6

附录 4 1999 年全国奥林匹克竞赛题选编

(162)	1.1.1
(169)	1.1.2
(178)	1.1.3
(192)	1.1.4
(202)	1.1.5
(213)	1.1.6

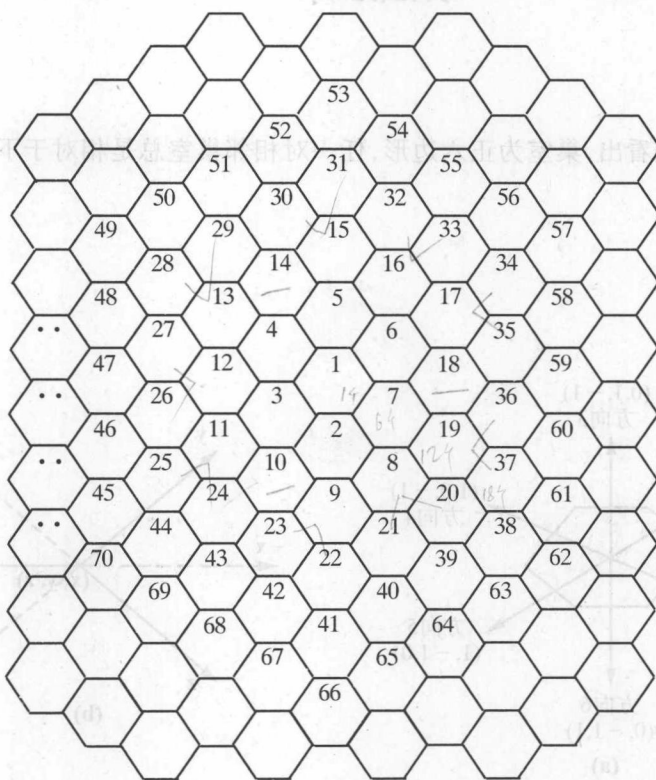
第一章 '99 国际大学生计算机程序设计竞赛 试题解析

§ 1.1 蜜蜂的繁殖

试 题

B. Heif 教授正在对一种他在巴西雨林中考察时发现的南美蜂进行实验,这种蜂能生产出比欧洲和北美蜜蜂更优质的蜜,不幸的是,这些蜂在饲养时繁殖得不好。Heif 教授认为原因是蜂窝中不同幼虫的放置依赖环境条件,而他实验室的环境条件和雨林不同。

作为证实他理论的第一步,Heif 教授想要定量地表示出幼虫放置的差异。要做到这一点,他得测量蜂窝中放置了幼虫的巢室(cell)间的距离。为了这个目的,教授用这种方法标记巢室:把任一巢室标为 1,然后把剩余巢室顺时针地标记,如(图 1.1-1)所示。



(图 1.1-1)

例如,两个在巢室 19 和 30 里的幼虫相距 5 个巢室,连接这两个巢室的最短路径之一为如下这些巢室组成的路径 19-7-6-5-15-30,所以你必须移动到相邻巢室五次,才能从 19 到 30。

Heif 教授需要你帮助他写一个程序,计算任何一对巢室间的距离,距离由最短路径上的巢室个数确定。

输入

输入由多行组成,每行含两个整数 a 和 $b(a, b \leq 10000)$,表示巢室编号。除了最后一行 $a = b = 0$ 以外,这两个整数总为正。这最后一行结束输入且不必处理。

输出

对于输入文件中的每一对数 (a, b) ,输出编号为 a 和 b 的巢室间的距离。距离为从 a 到 b 的最少移动次数。

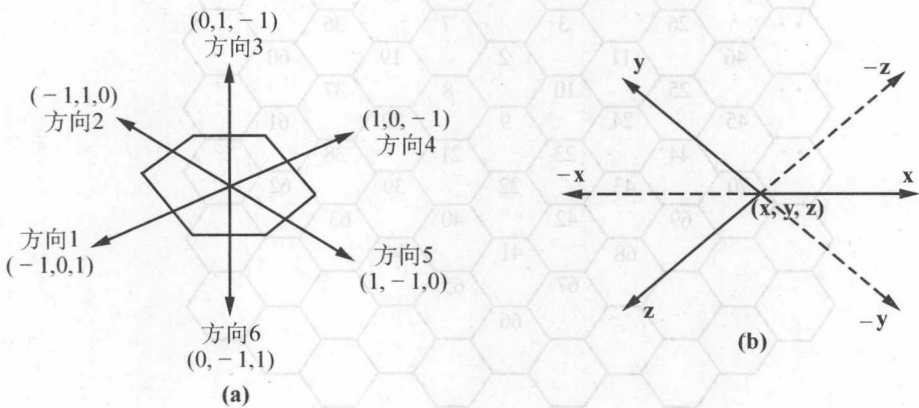
输入样本	输入样本的对应输出
19 30 0 0	The distance between cells 19 and 30 is 5.

算法分析

一、巢室的坐标

由(图 1.1-1)可以看出,巢室为正六边形,任一对相邻巢室总是相对于下述三个方向之一(图 1.1-2(a)):

- ①垂直方向;
- ②左斜线方向;
- ③右斜线方向。



(图 1.1-2)

我们将水平方向定义为 x 轴, 右为正, 左为负; 左斜线 $\swarrow$ 定义为 y 轴, 左上为正, 右下为负; 右斜线 $\searrow$ 定义为 z 轴, 左下为正, 右上为负(图 1.1-2(b))。一个巢室由坐标 (x, y, z) 表示, 其六个方向上相邻巢室的坐标增量分别由(图 1.1-2(a))中箭头一侧的三元组描述。设

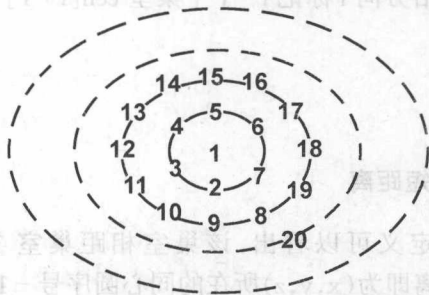
$\text{dmove}[i]$ — i 方向上的三个坐标增量($1 \leq i \leq 6$)。显然
 $\text{dmove}[1 \cdots 6] = ((-1, 0, 1), (-1, 1, 0), (0, 1, -1), (1, 0, -1), (1, -1, 0), (0, -1, 1))$
 由此得出 (x, y, z) i 方向的巢室 (x', y', z') 为

$$\begin{aligned} x' &= x + \text{dmove}[i, 1]; \\ y' &= y + \text{dmove}[i, 2]; \\ z' &= z + \text{dmove}[i, 3]; \end{aligned}$$

若两个巢室在 x 轴、 y 轴、 z 轴的增量分别为 $|\Delta x|$ 、 $|\Delta y|$ 、 $|\Delta z|$, 则这两个巢室相距的最短距离为 $\max\{|\Delta x|, |\Delta y|, |\Delta z|\}$ 。

二、标记蜂窝

我们给巢室 1 定义坐标 $(0, 0, 0)$, 并作为同心圆的圆心。巢室 1 正下方为巢室 2, 其坐标为 $(0, -1, 1)$, 由此出发, 顺时针方向标记巢室 3...巢室 7, 其坐标分别为 $(-1, 0, 1)$ 、 $(-1, 1, 0)$ 、 $(0, 1, -1)$ 、 $(1, 0, -1)$ 、 $(1, -1, 0)$, 巢室 2...巢室 7 构成以巢室 1 为圆心的第 2 个同心圆, 这 6 个巢室与巢室 1 的距离为 1 个单位长度; 巢室 7 正下方为巢室 8, 其坐标为 $(1, 0, -1 + (-1), 0 + 1)$, 由此出发, 顺时针标记巢室 9...巢室 19, 巢室 8...巢室 19 构成以巢室 1 为圆心的第 3 个同心圆, 这 12 个巢室与距巢室 1 的距离为 2 个单位长度。依次类推, 第 r 个同心圆上含 $6 * (r - 1)$ 个巢室, 这些巢室与巢室 1 的距离为 $(r - 1)$ 个单位长度。



(图 1.1-3)

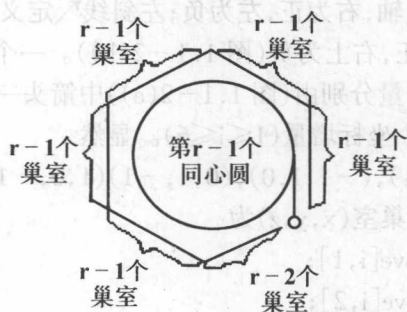
在标记第 r 个同心圆时, 按照顺时针方向要求, 首先在第 r 个同心圆最后一个巢室的正下方(方向 6)开始标记该圆上的第 1 个巢室, 接着沿方向 1 标记 r 2 个巢室, 使得方向 1 上含 $r - 1$ 个巢室; 然后再沿方向 2, 方向 3...方向 6 标记, 在每一个方向上顺序标志 $r - 1$ 个巢室。

设 $\text{cell}[l]$ —巢室 l 的三元坐标组。其中

$\text{cell}[l, 1]$ —巢室 l 的 x 坐标;
 $\text{cell}[l, 2]$ —巢室 l 的 y 坐标;
 $\text{cell}[l, 3]$ —巢室 l 的 z 坐标。

从巢室 1 出发, 沿 d 方向标记 k 个巢室的方法如下:

```
for i:=1 to k do
begin
    l←l+1; cell[l]←cell[l-1]d 方向的坐标;
```

(图 1.1-4)

end;

由此得出标记蜂窝的算法:

$l \leftarrow 1; \quad \text{cell}[1] \leftarrow (0,0,0);$

for $r := 2$ to 同心圆数的最大值 do

begin

$\text{cell}[l+1] \leftarrow \text{cell}[l]$ 方向 6 上的坐标; $l \leftarrow l+1;$

从巢室 l 出发沿方向 1 标记 $r-2$ 个巢室 $\text{cell}[l+1] \text{ cell}[l+r-2];$

$l \leftarrow l+r-2;$

for $i := 2$ to 6 do

begin

从巢室 l 出发,沿方向 i 标记 $r-1$ 个巢室 $\text{cell}[l+1] \text{ cell}[l+r-1];$

$l \leftarrow l+r-1;$

end; {for}

end; {for}

三、计算巢室 a 到巢室 b 的最短距离

从巢室坐标 (x, y, z) 的定义可以看出,该巢室相距巢室 1 $((0,0,0))$ 的最短距离为 $\max\{|x|, |y|, |z|\}$ 。这个距离即为 (x, y, z) 所在的同心圆序号 - 1。设巢室 a 和巢室 b 的坐标分别为 (x_a, y_a, z_a) 、 (x_b, y_b, z_b) 。我们通过坐标平移的办法将原点平移至巢室 b 的位置,使得巢室 a 的坐标变换至

$$x'_a = x_a - x_b; \quad y'_a = y_a - y_b; \quad z'_a = z_a - z_b$$

这样一来,便得出了巢室 a 和巢室 b 的最短距离, $\max\{|x_a - x_b|, |y_a - y_b|, |z_a - z_b|\}$, 即

$$\max_{1 \leq i \leq 3} \{|\text{cell}[a, i]| - |\text{cell}[b, i]|\}$$

程序题解

```
program bee _ breeding;
```

```
type
```

```
  tcoordinate = array [1..3] of integer;
```

{坐标类型}

begin

const

maxlabel = 10000; {巢室编号的最大值}

dmove: array [1..6] of tcoordinate = {dmove[i]—方向 i 上的三个坐标增量}

((-1,0,1), (-1,1,0), (0,1,-1), (1,0,-1), (1,-1,0), (0,-1,1));

var

cell: array [1..maxlabel] of tcoordinate; {cell[i]—巢室 i 的坐标}

a, b: integer;

procedure move(var c: tcoordinate; d: byte; var result: tcoordinate);

{输入坐标 c 和方向 d。计算和返回与 c 在 d 方向相邻的坐标 result}

begin

result[1] := c[1] + dmove[d][1];

result[2] := c[2] + dmove[d][2];

result[3] := c[3] + dmove[d][3];

end; {move}

function labelover(var l: integer; d, k: integer): boolean;

{从巢室 l 出发,沿 d 方向标记 k 个巢室}

var

i: integer;

begin

labelover := true;

for i := 1 to k do

begin

inc(l);

if l > maxlabel then exit;

move(cell[l-1], d, cell[l]);

end; {for}

labelover := false;

end; {labelover}

procedure labelcells;

var

r, l, i: integer;

c: tcoordinate;

begin

l := 1;

fillchar(cell, sizeof(cell), 0);

{从巢室 1 开始标记}

{cell 初始化}

```

for r := 2 to maxlabel do                                {顺序标记各个同心圆}
begin
  if labelover(1, 6, 1) then exit;                       {标记巢室1正下方的一个巢室}
  if labelover(1, 1, r-2) then exit;
  {沿方向1标记r-2个巢室,使得方向1上含r-1个巢室}
  for i := 2 to 6 do                                     {沿方向2...方向6标记,每个方向上顺序标记r-1个巢室}
    if labelover(1, i, r-1) then exit;
  end; {for}
end; {labelcells}

```

```

function getdistance(a, b: integer): integer; {计算和返回巢室a和巢室b的最短距离}
var
  result, i: integer;
begin
  result := 0;
  for i := 1 to 3 do
    if abs(cell[a][i] - cell[b][i]) > result then
      result := abs(cell[a][i] - cell[b][i]);
  getdistance := result;
end; {getdistance}

```

```

begin
  labelcells;                                           {标记蜂窝}
  assign(input, f'bees.ind');                          {输入文件名串与文件变量连接}
  assign(output, 'bees.out');                          {输出文件名串与文件变量连接}
  reset(input);                                         {输入文件读准备}
  rewrite(output);                                       {输出文件写准备}

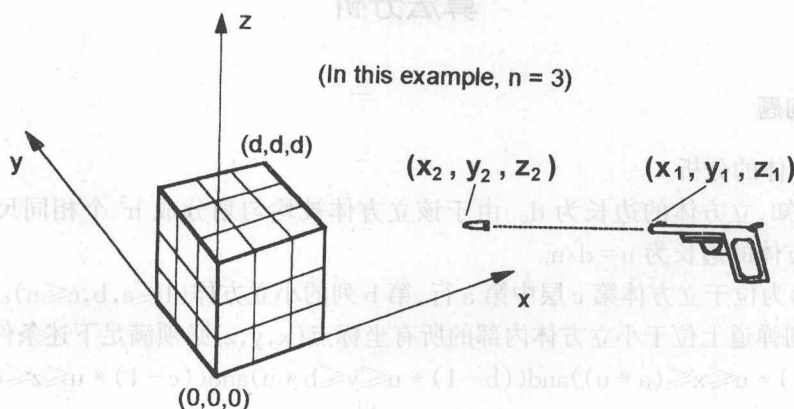
  readln(a, b);
  while not (a + b = 0) do                             {若文件未读完}
  begin
    writeln('the distance between cells ', a, ' and ', b, ' is ', getdistance(a, b), '.');
    {计算输出这一对巢室间的最短距离}
    readln(a, b);                                       {输入下一对巢室}
  end; {while}
end. {main}

```

§ 1.2 弹 孔

试 题

一个立方体悬在空间内。笛卡尔坐标系的原点在立方体的一个底角上,如图 1.2-1 所示。立方体边长为 d ,所以其对角在坐标 $(0, 0, 0)$ 和 (d, d, d) 处。坐标系的 z 轴正方向为相对于重力向“上”。



(图 1.2-1)

立方体的内部在每一维上被均匀划分,使得立方体被划分为 n^3 个相同尺寸的小立方体。隔板薄且不透水,同时每个小立方体都被注满水。所有小立方体中水的总体积为 d_3 。

一支枪射出一发子弹,可能击中立方体。枪口在点 (x_1, y_1, z_1) 处,点 (x_2, y_2, z_2) 是弹道上一点,由此确定子弹的方向。子弹没有击碎立方体,而是每当子弹击中立方体的一面或内部的隔板时,子弹留下一个小的弹孔。弹孔可能在内部小立方体的面、边或角上。受重力影响,水可能在这些小弹孔间渗漏,所有漏出大立方体的水均被收集,测量。

输入

输入数据组由多次实验组成,每次实验由八个整数描述,第一个整数是 n ($n \leq 50$),第二个整数是 d ($d \leq 100$);剩下的六个整数 $x_1, y_1, z_1, x_2, y_2, z_2$ 表示弹道起点和弹道上的某一点 ($-100 \leq x_1, y_1, z_1, x_2, y_2, z_2 \leq 100$)。弹道起点和弹道上的某一点不相同。最后一次实验之后,整数 0 结束输入数据组。

输出

你的程序必须计算出漏出大立方体的水的总体积。对于每次实验,打印实验编号,记号 Volume = ,和精确到小数点后两位的水的总体积。在每两次实验之间打印一个空行。

注意

在这道题中,如果两个实数之差小于 10^{-6} ,则这两个实数被认为是相等的。

输入样本

输入样本的对应输出

5	25	5	15	0	5	15	100	Trial 1, Volume = 2500.00
3	30	0	-35	0	3	-25	3	Trial 2, Volume = 1950.00
10	16	8	17	11	12	19	6	Trial 3, Volume = 0.00
0								

算法分析

一、明确几个问题

1. 小立方体的分析

由试题可知,立方体的边长为 d 。由于该立方体被均匀划分成 n^3 个相同尺寸的小立方体,因此小立方体的边长为 $u=d/n$ 。

设 (a,b,c) 为位于立方体第 c 层中第 a 行、第 b 列的小立方体 $(1\leq a,b,c\leq n)$ 。若子弹经过该小立方体,则弹道上位于小立方体内部的所有坐标点 (x,y,z) 必须满足下述条件

$$((a-1)*u\leq x\leq (a*u))\text{and}((b-1)*u\leq y\leq (b*u))\text{and}((c-1)*u\leq z\leq (c*u))$$

(公式 1)

2. 计算弹道与平面的交点

空间上的两点可确定一条直线。设弹道经过点 (x_1,y_1,z_1) 和点 (x_2,y_2,z_2) 。与之相交的三个平面分别为

$Z=z$,即该平面与 xoy 平面平行;

$X=x$,即该平面与 yoZ 平面平行;

$Y=y$,即该平面与 xoz 平面平行;

弹道穿过 $Z=z$ 平面的弹孔坐标为

$$x_{z=z}=\frac{x_1(z-z_2)-x_2(z-z_1)}{z_1-z_2}$$
$$y_{z=z}=\frac{y_1(z-z_2)-y_2(z-z_1)}{z_1-z_2}$$
$$z_z=z$$

(公式 2)

在计算小立方体上表面和下表面的弹孔位置时,使用上述公式。

弹道穿过 $Y=y$ 平面的弹孔坐标为

$$x_{y=y}=\frac{x_1(y-y_2)-x_2(y-y_1)}{y_1-y_2}$$
$$y_y=y$$
$$z_{y=y}=\frac{z_1(y-y_2)-z_2(y-y_1)}{y_1-y_2}$$

(公式 3)

在计算小立方体前表面和后表面的弹孔位置时,使用上述公式。

弹道穿过 $X=x$ 平面的弹孔坐标为

$$\begin{aligned} x_{x=x} &= x \\ y_{x=x} &= \frac{y_1(x-x_2) - y_2(x-x_1)}{x_1 - x_2} \\ z_{x=x} &= \frac{z_1(x-x_2) - z_2(x-x_1)}{x_1 - x_2} \end{aligned} \quad (\text{公式 4})$$

在计算小立方体左表面和右表面的弹孔位置时,使用上述公式。

3. 判断弹道是否穿过立方体

由 (x_1, y_1, z_1) 和 (x_2, y_2, z_2) 确定的弹道若满足下述 6 个条件中的任意一个或多个,则该弹道不能穿过边长为 d 的立方体

$d < x_1 < x_2$	弹道位于立方体左方
$x_2 < x_1 < 0$	弹道位于立方体右方
$d < y_1 < y_2$	弹道位于立方体后方
$y_2 < y_1 < 0$	弹道位于立方体前方
$d < z_1 < z_2$	弹道位于立方体上方
$z_2 < z_1 < 0$	弹道位于立方体下方

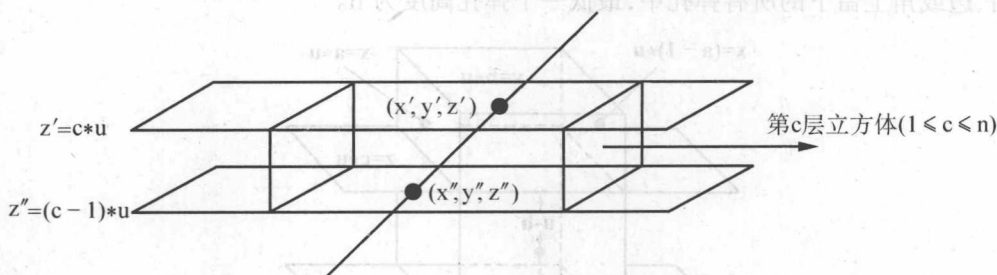
换句话说,若上述 6 个条件同时不满足,即

$$(\overline{d < x_1 < x_2}) \text{ and } (\overline{x_2 < x_1 < 0}) \text{ and } (\overline{d < y_1 < y_2}) \text{ and } (\overline{y_2 < y_1 < 0}) \text{ and } (\overline{d < z_1 < z_2}) \text{ and } (\overline{z_2 < z_1 < 0}) \quad (\text{公式 5})$$

则弹道穿过立方体。

二、自上而下,逐层分析弹道轨迹

在确定弹道穿过立方体的前提下,按自上而下的方向,一层一层地分析立方体,每一层的 $n \times n$ 个小立方体与 xoy 平面平行。在分析第 c 层立方体时,根据(公式 2)分别计算弹道与平面 $z' = (c-1) * u$ 和平面 $z = c * u$ 的两个交点 (x'', y'', z'') 、 (x', y', z') 。



(图 1.2-2)

1. 计算弹道经过立方体第 c 层中的哪些小立方体

设弹道经过第 $c (1 \leq c \leq n)$ 层时,穿过一个由 $a'' - a'$ 行和 $b'' - b'$ 列的小立方体组成的子平面。那么如何计算这个子平面的左下角坐标 (a'', b'') 和右上角坐标 (a', b') 呢?

首先分析弹道方向。如果子弹沿平行于 xoy 平面的方向 $(z_1 = z_2)$ 飞来,并且 $(c-1) * u \leq z_1 \leq c * u$, 则 $x' = x_1, y' = y_1, x'' = x_2, y'' = y_2$ 。