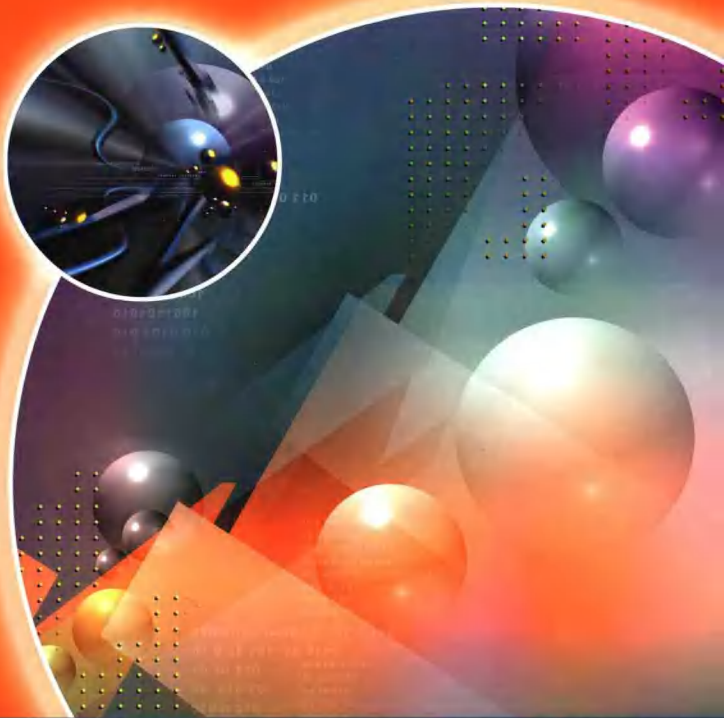


经全国中小学教材审定委员会 2004 年初审通过

普通高中课程标准实验教科书

算法与程序设计

(信息技术·选修1)



PUTONGGAOZHONG KECHENGBIAOZHUN SHIYANJIAOKESHU

SUANFAYUCHENGXUSHEJI

PUTONGGAOZHONG KECHENGBIAOZHUN SHIYANJIAOKESHU
SUANFAYUCHENGXUSHEJI

普通高中课程标准实验教科书

算法与程序设计



普通高中课程标准实验教科书

算法与程序设计

陶增乐 主编

- ▶ 出版发行 浙江教育出版社
(杭州市天目山路40号 邮编:310013)
- ▶ 责任编辑 华 明
- ▶ 装帧设计 韩 波
- ▶ 责任校对 雷 琴
- ▶ 责任印务 温劲风
- ▶ 图文制作 杭州兴邦电子印务有限公司
印 刷 浙江印刷集团有限公司

- ▶ 开 本 890×1240 1/16
- ▶ 印 张 8.75
- ▶ 字 数 280 000
- ▶ 版 次 2005年7月第1版
- ▶ 印 次 2006年6月第3次
- ▶ 书 号 ISBN 7-5338-5944-8/G·5914
- ▶ 定 价 10.08元

ISBN 7-5338-5944-8



9 787533 859442

批准文号:鄂价[2006]138号 邮购电话:12338
联系电话:0571-85170300-80828
e-mail:zjpe@zjpe.com
网址: www.zjpeh.com

编者的话

一提起“算法”，人们立刻会觉得这是一门高深的学问，难度很大。其实，这是一种误解。在日常生活中，我们时时处处都在与“算法”打交道。舞台上舞步的图谱，是舞蹈演员表演舞蹈的算法；乐谱是乐队演奏和指挥的算法；编织毛衣生成各种图案花纹的针法是手工编织师的算法；菜谱是厨师烧菜的算法。我们学习信息技术，与计算机打交道，用计算机写文章，上网浏览、发E-mail，甚至单击一个按钮、选取一项菜单等等，在计算机内部都是通过一系列“计算”的结果，只不过软件设计者为方便大家的使用，煞费苦心地将算法“隐藏”起来，人们看见的只是通过一些简单操作便能获得所需要的一些结果。其实，以上种种尽管在各种场合有自己独特的称呼，但它们都具有“算法”的共同特点，即把工作变成一系列机械步骤，这些工作都有一个起始步，每一步的后继步都是明确的，经过有限个步骤结束工作。如果问题解决，就叫做算法正确；如果问题不能解决，就是没有找到正确算法或问题本身不存在可行的算法。由上可知，不论是日常生活，还是信息技术，人们到处用到“算法”，“算法”无处不在，只不过通常不这样称呼罢了。

那么“算法”到底难不难呢？首先，算法是一种问题解决方法的机械化，把解法方法变成一个明确无误的有限步骤序列。通常，求解简单问题的算法也相对简单些，解决复杂问题的算法会相对复杂些。但这不是绝对的，对于解决同一问题可以设计出各种不同的算法，一个算法的好坏主要取决于它需要耗费多少计算时间和占用多少存储空间。

首先，在高中的计算机课程中，曾经试验过以程序设计语言教学为中心的做法，事实上，算法才是程序设计的依据，具体程序只是算法的一种描述手段。为此，我们花了相当多的篇幅，以问题解决为核心，用较易理解的自然语言和流程图语言来描述算法（在本书所附光盘中有书中全部实例的算法执行过程的流程图演示，形象生动而直观），让大家充分体验算法的作用，并逐步建立算法思维的方法。当然，算法最终必须用程序设计语言编写成程序才能在计算机上实现。本教材选用Visual BASIC语言来描述算法是因为它是一个可视化的面向对象的程序设计语言，在近几年各地信息技术课程教学中已被广泛采用。

这是一门选修课程，是信息技术学习的核心内容之一。通过本模块的学习，使大家进一步了解算法和程序设计在解决问题过程中的地位和作用，培养自己的算法思维能力；能从简单问题出发，设计解决问题的算法，并能初步学会使用一种程序设计语言编制程序实现算法以解决问题。希望大家结合其他课程的学习，充分发挥创造性，把“算法”学习得生动活泼。

最后，对大家的学习提一点建议。在日常生活、学习中，我们经常用食青果来比喻“苦尽甘来”。青果，入口苦涩而酸，然久嚼，回味却清甜生津，馥郁留香。学习算法亦如此，只要有耐心，锲而不舍，不怕困难，一定会回味无穷。我们是过来之人，现在还在努力学习算法，说这些话的目的是为了互勉。

编者
2005年6月

高级中学课程标准实验教科书编写人员

主 编：陶增乐

副 主 编：吴洪来 王荣良

主要撰稿人：周建中 吴申广

责任编辑：华 明

装帧设计：韩 波



目 录

SUANFAYUCHENGXUSHEJI

第一章 算法和算法的表示	1
1.1 使用计算机解决问题的一般过程	2
1.2 确定解决问题的方法	3
1.3 把解决问题的方法步骤化	4
1.4 算法的概念和表示方法	7
本章小结	18
第二章 算法实例	20
2.1 枚举算法	21
2.2 解析算法	25
2.3 排序	30
2.4 查找	39
本章小结	45
第三章 面向对象程序设计的基本知识	46
3.1 面向对象程序设计方法简介	47
3.2 在可视化的程序设计环境 VB 中建立一个应用程序	55
本章小结	66
第四章 VB 程序设计初步	68
4.1 基本数据类型、常量和变量	69
4.2 基本运算和表达式	72
4.3 语句	75
4.4 过程和函数	83
本章小结	91



目 录

SUANFAYUCHENGXUSHEJI

第五章 算法实例的程序实现	94
5.1 枚举算法的程序实现	95
5.2 解析算法的程序实现	101
5.3 排序算法的程序实现	107
5.4 查找算法的程序实现	111
5.5 递归算法实例及程序实现	116
本章小结	130
附录1 程序设计语言发展简史	132
附录2 学生学业成长记录表	133
部分中英文术语对照表	136



1.1 使用计算机解决问题的一般过程

随着科技的发展,计算机的性能日益提高,价格不断下降,越来越多的人使用计算机来解决各种各样的问题,以提高工作的效率。建筑师使用计算机来进行应力分析和结构设计,以获得建筑物设计方案中需要的各种数据,过去他们用纸和笔进行计算,需要花费几个月甚至更长的时间才能得到结果,而计算机往往只需要几分钟至多几小时就完成了。银行职员使用计算机来处理客户的存贷业务,他们只需要把交易数据输入计算机,由计算机自动地进行各种业务处理,每日自动地进行汇总,打印各种报表,再也不用银行业务人员拨弄算盘,抄录数据和填写报表。冰箱制造厂的工程师使用计算机,逐一地检测装配线上冰箱的性能,不合格的产品不可能逃过计算机的“眼睛”。教师和学生使用计算机,在因特网上获得教学资料,制作图文并茂的学习心得和各种实验报告,对一个阶段的学习进行总结交流。



知识链接

指令和指令系统

指令:用来规定计算机操作的命令。

指令系统:一台计算机全部指令的集合。

看来计算机似乎是无所不能的,实际上,至今为止计算机只能按照设计好的程序,一步一步地进行计算,只是由于计算机运行的速度非常快,使人们感到只要一按鼠标键就万事大吉了。事实上,要使用计算机解决某个问题,人们必须事先设计好解决这个问题的计算机程序,例如,要用计算机写一篇文章,就要有 Word 这样的字处理软件。当我们输入文字,插入图片或表格时,计算机将按照 Word 中设计好的处理这些问题的步骤,进行复杂的处理,使我们能得到符合各种风格要求的文本,如果没有解决这个问题的计算机程序,即使问题再简单,也不可能用计算机来解决这个问题。例如,解一元二次方程,这个初中学生都能解决的问题,如果事先未设计好计算它的程序,那么,计算机还是无法解决这个十分简单的问题的。

所谓用计算机来解决某个问题,实际上是用计算机来处理与这个问题相关的信息。现实世界中的问题种类繁多,有简单的,也有相当复杂的,要求计算机处理的信息,涉及面很广。

使用计算机解决问题一般要经历三个阶段:

1. 分析问题确定要用计算机做什么

这个阶段的主要目标是确定要计算机“做什么”。例如,要求 Word 能为我们编辑包含文字、图片、表格的文本,要计算机求解一元二次方程,即解出 $ax^2+bx+c=0$ 的实数根(或告诉我们无实数根),等等,即确定解题的任务。

2. 寻找解决问题的途径和方法

接下来,是要解决“怎样做”的问题。

至今为止,人们已经开发、积累了丰富的解决各种问题的算法并编写出相应的计算机程序。我们可以根据第一阶段确定的任务,寻找、挑选、购买合适的解题应用程序或应用软件,来解决我们的问题。

由于现实世界中问题的多样性,并不是所有的问题都有现成的计算机程序可用的。在这种情况下,我们必须根据问题求解的实际任务,自己动手设计解决问题的算法。

3. 用计算机进行处理

如果我们购买了现成的、合适的计算机软件,那么,就可以按照这个软件的使用方法,进行适当的操作,例如,输入问题的原始数据,命令计算机执行相应的处理,以得到所需要的计算结果。



如果没有现成的软件可用,就需自己设计算法,然后使用适当的程序设计语言来编制一个实现这一算法的计算机程序。根据问题所给的数据和所设计的算法,用程序描述怎样从原始数据出发,一步一步地执行算法,以获得问题所需的计算结果。

使用计算机来解决问题,算法是重要的。实际上,在现成的软件中,例如在 Word 中,就包含了许多比较复杂的算法。我们自己设计一个程序来解决特定的问题时,同样需要寻找或设计解题的算法。

1.2 确定解决问题的方法

下面将通过若干实例来进一步说明用计算机解决实际问题的步骤。

例 1 使用一根长度为 L 厘米的铁丝,制作一个面积为 S 平方厘米的矩形框,要求计算该矩形框应有的高 h 和宽 w 。这里,铁丝的长度 L 和矩形框面积 S 是根据需要预先指定的,如图 1.2.1 所示。

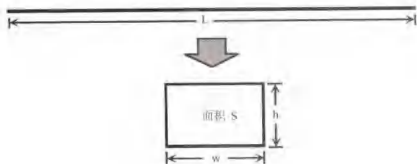


图 1.2.1 用铁丝制作一个矩形框

[方法]

设矩形框的高为 h 厘米,可知宽 $w = \frac{L}{2} - h$ 厘米,则

$$h \left(\frac{L}{2} - h \right) = S,$$

整理后得到下面的一元二次方程

$$h^2 - \frac{Lh}{2} + S = 0,$$

对于这个一元二次方程可以使用求根公式

$$h_1, h_2 = \frac{L \pm \sqrt{L^2 - 16S}}{4},$$

计算方程的根。设 $d = L^2 - 16S$, 则

当 $d > 0$ 时, 方程的一个根是矩形框的宽

$$w = \frac{L + \sqrt{L^2 - 16S}}{4},$$

另一个根就是矩形框的高

$$h = \frac{L - \sqrt{L^2 - 16S}}{4};$$



当 $d=0$ 时,方程有两个相同的根 $\frac{L}{4}$,即矩形框的高和宽相等,是一个边长为 $\frac{L}{4}$ 的正方形;

当 $d<0$ 时,方程无实数根,这种情况下不能制作出所要求的矩形框。



问题与思考

- 你平日在解决生活、学习中的问题时,通常用什么方法来表述解决问题的过程和结果,举例说明之。
- 你在学习、生活中遇到的问题是否都能表示成数学公式?
 - 列举几个能表示成数学公式的问题的例子。
 - 列举几个不能表示成数学公式的问题的例子,并说明你的解决步骤。



知识链接

程序

为解决某一问题而编排的一个指令序列称为程序(program)。

1.3 把解决问题的方法步骤化

为使计算机能按照上面确定的方法进行计算,光有计算公式是不够的,还必须把解决问题的方法步骤化,即要用某种方式告诉计算机,第1步做什么,第2步做什么,一般地,第*i*步做什么,第*i+1*步做什么。

对于上面确定的方法而言,在进行计算之前,计算机必须要知道给定的铁丝长度 L 和所要求的矩形框面积 S 。即,要把计算所需的原始数据 L 和 S 输入到计算机中,然后才能按确定的公式一步一步地进行计算,并输出最终的计算结果。

计算机开始计算之前,必须把解决这一问题的程序存储在计算机的内存中。通常,一个程序由如下两部分组成:

(1) 指令部分:指令是对计算机操作类型和操作数地址作出规定的一组符号。指令部分由一系列的指令构成,每条指令指定了要求计算机应执行的一个动作。由适当的指令构成的指令序列,描述了解决这个问题的计算过程。

(2) 数据部分:计算所需的原始数据、计算的中间结果或最终结果。

下图是一个示意性的程序 P ,用来说明计算矩形框的高度和宽度的步骤。

图中的一个方块代表内存中的一个存储单元,在每个存储单元中可以存储一条指令或一个数据。每个存储单元都有一个唯一确定的序号,标在每个方块的左边。



[示意性程序 P]

	序号	存储单元的内容
指令区	1	显示文字“请输入长度。”
	2	接收输入数据(铁丝的长度)送到变量 L
	3	显示文字“请输入面积。”
	4	接收输入数据(矩形框的面积)送到变量 S
	5	计算 L^2-16S , 结果送变量 d
	6	若 $d=0$ 转到 12
	7	若 $d<0$ 转到 15
	8	输出文字“两条不同的边长”
	9	输出: $(L+\sqrt{d})/4$
	10	输出: $(L-\sqrt{d})/4$
	11	结束
	12	输出文字“两条相同的边长”
	13	输出: $L/4$
	14	结束
	15	输出文字“无解”
	16	结束
数据区	1	变量 L 中的数值
	2	变量 S 中的数值
	3	变量 d 中的数值

读这个示意性的程序 P 时应注意:

(1) 从指令区序号为 1 的内存单元开始,按顺序一条一条执行程序 P 中的指令。

(2) 在执行序号为 6 的单元中的指令时,如果 $d=0$,则转到序号为 12 的内存单元,即当变量 d 中的数据为 0 时,就改变指令执行的次序,使序号为 12 的内存单元中的指令,成为下一条将要执行的指令;如果变量 d 中的数据不等于 0,则接着执行序号为 7 的单元中的指令。

(3) 任何时候,只要执行了“结束”指令,整个程序执行过程就被终止了。

对于计算机而言,程序是非常重要的,计算机的工作就是忠实地执行程序,有什么样的程序,计算机就做什么样的工作。计算机能处理图文并茂的文稿的编辑工作,是因为计算机的内存中装载并运行了文字处理程序;我们能在因特网上进行浏览,是因为计算机运行着一个叫做浏览器的程序。如果在计算机的内存中装载并运行解例 1 的程序 P,则计算机就成为求解例 1 问题的工具。

为了解决不同的问题,人们必须设计不同的程序。设计一个程序时,需要考虑以下问题。

1. 数据的存储

计算所需要的原始数据(如例 1 中铁丝的长度 L 和矩形框的面积 S)、计算产生的中间结果(如 L^2-16S)需要存储在不同的变量中(如示例程序 P 中的变量 L, S, d)。



2. 计算的过程

首先必须确定解决问题的方法,接着要把该方法步骤化,并用计算机能执行的指令来实现对应的步骤。由这些指令所组成的指令序列构成了一个程序,它会被计算机用来解决这一问题。也就是说程序描述了问题的计算过程。这意味着程序中不仅必须指出计算机应做的动作(例如,从哪里取得数据,进行哪些计算后,又将所得的结果存储到哪里),也必须指出动作的次序(例如,执行完当前这条指令后,下一条将要执行的指令是什么)。

一台计算机可以使用的指令种类是有限的,典型的指令类型有:

输入指令:通过输入设备(例如键盘、鼠标等),向程序输入数据,并存储到指定的变量中。

输出指令:把计算获得的结果,通过输出设备(例如显示器或打印机)输出。

算术运算指令:进行加、减、乘、除等算术运算。

逻辑判断指令:可以对指定的两个数据进行大小(或相等性)比较,比较的结果将产生一个逻辑值(真或假)。也可以使用逻辑运算(例如与、或、非),把若干个较简单的判断连接起来,形成一个复杂的判断。

控制转移指令:用来改变程序中指令的执行顺序。通常,程序中的指令是一条一条顺序执行的;但有时可以根据算法的需要,改变指令执行的顺序。

阅读材料



算法的学习和研究是计算机科学的基石

在使用计算机完成任何一项任务之前,必须找到能完成该任务的算法,并使用与该机器兼容的编程语言表示这一算法,用程序设计语言表示的算法称为程序。

算法研究开始时是作为数学学科的一个分支出现的,在现代电子数字计算机产生之前的一个很长时期内,算法的研究是数学家的一项重要研究内容,它的目的是寻求描述解决某一特定类型的所有问题的一组命令。早期算法研究的著名例子是求两个多位数的商的大除算法,以及求两个正整数的最大公因子的算法,都是由古希腊的数学家欧几里得(Euclid)发现的。

完成一个任务的算法一旦被找到,在完成该任务时就不要遵循算法所指定的步骤执行下去,而不再需要了解该算法所根据的原理了。例如,人们可以根据欧几里得算法来求出两个正整数的最大公因子,但并不需要了解这个算法为什么是这样的,在某种程度上说,解决这类问题的智能已经被编码成这一算法。

通过算法来获得和表达这智能,至少是智能行为,人们能够建造起完成各种任务的机器,因此,机器所表现出的智能水平受到算法所表达的智能的限制,仅当人们找到了完成某种任务的算法之后,才可能建造完成该任务的机器。反之,如果找不到解决这一问题的算法,那么求解该问题就超出了机器的能力范围。

算法能力的局限性曾经作为数学的一个基本课题来加以研究。20世纪30年代出生于奥地利的数学家库尔特·哥德尔(Kurt Godel)发表了不完备性定理的论文,这个定理从本质上证明,任何一个包括传统意义的算术系统的数学理论领域内,存在着大量的既不能证实又不能否认的不可判定的命题。简言之,仅靠算法的能力并不能解决算术系统的全部问题,也就是说,对于算术系统的任何彻底研究已经超出了算法活动的能力。

这一结果给数学大厦的基础撕开了一条大大的裂缝,但是,关于算法功能的研究却导致了今天的计算机科学的产生。的确,算法的研究是今天的计算机科学的不可动摇的基础。



问题与练习

1. 试写一个求 $1-2+3-4+5-\cdots+(-1)^{n-1}n$ 的示意性程序(参照 P 程序格式,假定 $n=1000$)。
2. 试写一个求 1 到 100 之间的偶数之和的示意性程序(参照 P 程序格式)。

1.4 算法的概念和表示方法

用计算机解题时,任何答案的获得都是按指定顺序执行一系列指令的结果,因此,用计算机解题前,需要将解题方法转换成一系列具体的、在计算机上可执行的步骤,然后,才能将这些步骤表示成指令代码。这些步骤能清楚地反映解题方法一步步“怎样做”的过程,这个过程就是通常所说的算法。本节的任务是进一步阐明算法的基本概念和表达算法的方法。

1.4.1 算法的概念

“算法”(algorithm)一词是从 9 世纪阿拉伯数学家花拉子米(alKhwarizmi)的名字派生而来的。这位数学家的一生发现了许多求解算术问题的方法,并写了一本叫作《合并与回代》(aljabr w'almuqabalah)的书。合并与回代这两个词是指解方程时所用的两个主要过程。这本书后来翻译成了拉丁文,书名被简化成现在人们所熟悉的“代数学”(algebra)。

从上面解例 1 的程序 P 中,我们可以看到一个算法应该具有如下的特征:

(1) 有穷性。一个算法必须保证它的执行步骤是有限的,即它是能终止的。也就是说操作步骤不能是无限的。例如,在解例 1 的过程中,不论方程有一个根、两个根或者无根,最多只需要执行 11 个操作步骤,算法的执行便能终止。算法中可以有重复执行的步骤,只要这些步骤的执行能终止。广义地说,“有穷性”一般指操作步骤的数量有限或能在合理的时间范围内完成全部操作。有些算法,虽然是有穷的,但它所花费的时间如果超出合理的限度,例如,需要一台或一组现代高速计算机运行几十甚至几百年才能得到结果,那么这种算法,也不能算是有效的算法。

(2) 确定性。算法中的每个步骤必须有确切的含义,而不应当是含糊的,模棱两可的。例如,步骤

输出: $L / \text{正整数}$

是无法执行的,因为没有指定 L 除以哪一个正整数,所以,这个步骤是不确定的。

(3) 可行性。算法中的每一个步骤都要足够简单,是实际能做的,而且能在有限的时间内完成。例如,程序 P 中序号为 9 的单元中指令

输出: $(L+\sqrt{d})/4$

是在条件 $d>0$ 的前提下才被执行的,式中的求正数的平方根、求和以及计算两个正数的商,所有这些计算都能在非常短的时间内完成。

(4) 有 0 个或多个输入。所谓输入是指算法在执行时需要从外界获得



知识链接

算 法

图灵奖获得者 D.E.Knuth 在他的巨著《计算机程序设计技巧》第一卷中说:“算法的现代意义十分类似于处方、过程、方法、技术、规程、程序,只不过是算法一词的含义有时稍微有一点特殊。一个算法,就是一个有穷规则的集合,其中之规则规定了一个解决某一特定类型的问题的运算序列。”



数据,其目的是为算法建立某些初始状态。如果建立初始状态所需的数据已经包含在算法中了,那就不再需要输入。例如,解例1的算法需要输入两个数据(铁丝的长度 L 和矩形框的面积 S)。

(5)有一个或多个输出。算法的目的是用来求解问题,问题求解的结果应以一定的方式输出。例如,求解例1的结果为:

$$\begin{aligned} \text{两条不同的边长: 宽度} &= \frac{L + \sqrt{L^2 - 16S}}{4}, \\ \text{高度} &= \frac{L - \sqrt{L^2 - 16S}}{4} \quad (\text{两个输出}); \end{aligned}$$

$$\text{两条相同的边长: 宽度} = \text{高度} = \frac{L}{4} \quad (\text{一个输出});$$

$$\text{无解} \quad (\text{一个输出}).$$

“无解”告诉我们在一定的情况下($d < 0$)该问题没有解,这也是结果,也是输出的信息。没有输出的算法是毫无意义的。

知识链接

高纳德

高纳德(D.E.Knuth)是美国国家科学院院士,他的匠心之作《计算机程序设计技巧》,广泛、深入地探讨了构成计算机科学的两大主题:算法和程序设计语言,赢得了计算机学术界的一片赞誉。正是由于这部计算机科学的经典著作,使他在1974年获计算机科学领域的最高奖——图灵奖。高纳德还是计算机排版 TeX 技术的发明者。

事实上,在日常生活中解决问题经常要用算法,只是通常不用算法这个词罢了。例如,乐谱是乐队指挥和演奏的算法;菜谱是厨师做菜的算法等等。在漫长的岁月中,人们发现了很多算法。例如欧几里得提出的求两个自然数的最大公约数算法,早期希腊学者埃拉多塞尼(Eratosthenes)发现的寻找素数的筛法等都是著名的算法例子。电子计算机的出现,开创了算法研究的新时代。人们可以将算法编写成程序提交给计算机执行,从而迅速获得解题结果。著名计算机科学家高纳德(D.E.Knuth)认为:计算机科学是算法的科学。

1.4.2 算法的表示

一个算法可以用多种不同的方法来描述,流程图(flowchart)是一种比较直观易用的,用图形来描述算法的方法。

例2 学校上体育课,一般在操场上课,遇到下雨或下雪天,改到室内上课,可用流程图表示如下,见图1.4.1。

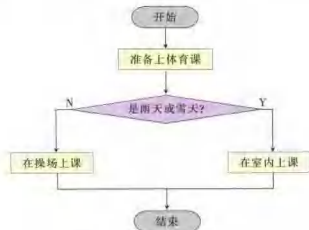


图 1.4.1 描述体育课流程图



流程图中所用的符号,应遵循国家颁布的标准(GB1526—89,ISO 5807—1985)来画。这套标准中最基本、最常用的构件有:

- ①处理框();框中指出要处理的内容,该框有一个入口和一个出口。
- ②输入输出框();用来表示数据的输入或计算结果的输出。
- ③判断框();用来表示分支情况,菱形框的四个顶点中,通常用上方的顶点表示入口,视需要用其余三个顶点来表示出口。
- ④连接框();用于连接因画不下而断开的流程线。
- ⑤流程线(\rightarrow);指出流程控制方向,即动作的次序。
- ⑥开始、结束符();用来表示算法的开始和结束。一个算法只能有一个开始处,但可以有多个结束处。

图 1.4.2 是用流程图表示的解例 1 问题的算法。

算法从“开始”处执行,依次执行了前四个输入框内的动作后,使用者已把计算所需的原始数据(铁丝的长度和矩形的面积)输入,并分别存储到程序的变量 L 和 S 中。第五个框是一个处理框,框内的动作

$$d \leftarrow L^2 - 16S$$

表示计算 $L^2 - 16S$ 的值,计算的结果存入变量 d 中。

接着执行的是由判断框表示的选择动作(即分支);取出变量 d 中的数据,若这个数据的值为 0,则沿着标有 Y(Yes)的有向线段,执行该有向线段所指向的动作(输出文字“两条相同的边长”);若变量 d 中取出的数据不为 0,则沿着标有 N(No)的有向线段,执行该有向线段所指向的动作(判断 $d < 0$ 是否为真)。

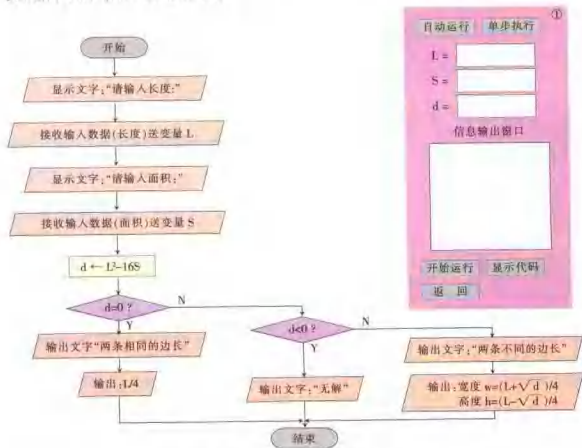


图 1.4.2 用流程图描述解例 1 的算法

① 请将配套光盘中的有关算法辅助程序复制到“运行体验”文件夹中,并按“算法执行过程体验”的要求,上机操作并回答指定的问题。本章以及第二章中的算法执行过程体验,均按这样的方式进行。



算法执行过程体验

双击“运行体验”文件夹中“求矩形框的边长”程序,完成下列操作。

1. 采用“自动运行”模式,按提示分别输入L的值为90,S的值为270,则输出结果为_____。
2. 继续采用“自动运行”模式,分别输入L的值为30,S的值为100,则输出结果为_____。
3. 若输入L的值为40,S的值为100,最后输出的结果为_____,共运行了_____步操作。

我们也可以使用自然语言,例如汉语,加上一些必要的数学符号,来描述解决问题的算法,如上面图1.4.1描述体育课问题。下面是用自然语言描述的解例1的算法。

- (1) (输入原始数据)输入:铁丝的长度送变量L。
矩形框的面积送变量S。
- (2) (计算d)计算 L^2-16S ,结果送变量d。
- (3) (判断是否仅有一个根)如果 $d=0$,那么转到7。
- (4) (判断是否无根)如果 $d<0$,那么转到9。
- (5) (方程有两个根)输出:“两条不同的边长”
宽度 $w=(L+\sqrt{d})/4$
高度 $h=(L-\sqrt{d})/4$
- (6) (算法终止)结束。
- (7) (方程有一个根)输出:“两条相同的边长”
宽度 $w=L/4$ 高度 $h=L/4$
- (8) (算法终止)结束。
- (9) (方程无实数根)输出:“无解”。
- (10) (算法终止)结束。

自然语言的主要缺点是有时会存在两义性,这种两义性不是算法本身的问题,是算法表示的工具引起的。例如,“打死老虎”,既可以表示“打的是一只死老虎”,也可以表示“打死了一只老虎”,因此它可能引起某些计算步骤的不确定性。

此外,我们还可以用“伪代码”(pseudo-code)来描述算法,伪代码使用某些程序设计语言中的控制结构,来描述算法中各步骤的执行次序和模式,使用自然语言、数学符号或其他符号,来表示计算步骤要完成的处理或需要涉及的数据。使用伪代码可以免去许多绘图的麻烦,但前提是必须熟悉某种程序设计语言。下面是用伪代码描述的解例1的算法。

知识链接

伪代码

伪代码是一种在算法开发过程中用来表达思想的非形式化的符号系统。相对编程语言来说,它对语法规则、语义结构等规定限制比较宽松,是一种更加直观易用的表示系统。

例如,前面提到的上体育课情况,可用如下形式表示 IF (未下雨或雾) THEN (在操场上课) ELSE (在室内上课)。

总之,伪代码的目的就是要提供一种可读性好、非形式化的语法符号规则。



```
read ( L ← 铁丝的长度, S ← 矩形框的面积 )
d ← L2-16S
if ( d=0 ) write ( “两条相同的边长”, L/4 )
else if ( d<0 ) write ( “无解” )
else write ( “两条不同的边长”, (L+√d)/4, (L-√d)/4 )
```

1.4.3 变量和变量的用途

程序中的变量和数学公式中的变量是有区别的。程序中的变量指的是数据的存储单元,其中存储的数据在程序的执行过程中是可变的。通过输入指令的执行,程序将外界输入的数据存储到指定的变量中,程序计算的结果也可以存储到指定的变量中。一旦把数据存储到某个变量,例如变量 v 中,只要我们不把新的数据送到变量 v ,那么,在程序整个运行过程中变量 v 将永久地保存着这个数据。如果在计算过程中需要使用 v 中的数据,我们可以从变量 v 中取出这个数据。注意,单是读取是不会改变变量的内容的。在大多数程序设计语言中,在定义变量时,都要明确说明变量的数据类型。

在一个问题的计算过程中,可能需要使用多个变量,来保存计算过程中要用到的数据,我们应该为每个变量指定一个适当的名称。

例 3 设计一个算法,计算某单位内所有职工的平均年龄。

实际上,这是一个计算一批数据的算术平均值的问题。这批数据由使用者从键盘输入,可不必预先指定数据的个数,并约定:输入 0 时表示本次计算所需的全部数据已输入完毕。

计算数据的算术平均值,是用途广泛的一种计算。例如,统计某个学校某年级的体育成绩,统计某公司一个季度的经营业绩等等。

首先考虑该问题中涉及的数据,设计适当的变量来保存这些数据:

d: 用来存储从键盘输入的一个数据,当输入的数字为 0 时,表示输入结束。

c: 用来记录已经输入的有效(非 0)数据的个数。

sum: 用来计算有效数据之和。

下面考虑计算的过程。为了计算一组数据的平均值,必须知道这批数据的个数 c ,以及所有这些数据之和 sum 。显然,我们需要的计算结果是 sum/c 。因此,我们首先应该通过输入指令,把使用者输入的数据(也可能是表示结束的 0)送到变量 d 中。如果变量 d 中接收到的是一个非零的数据,这个数据应该被累加到变量 sum 中,同时,使变量 c 的值增加 1,表示接收到了一个新数据。

图 1.4.3 是计算数据算术平均值的算法。