

中学信息技术
创新丛书

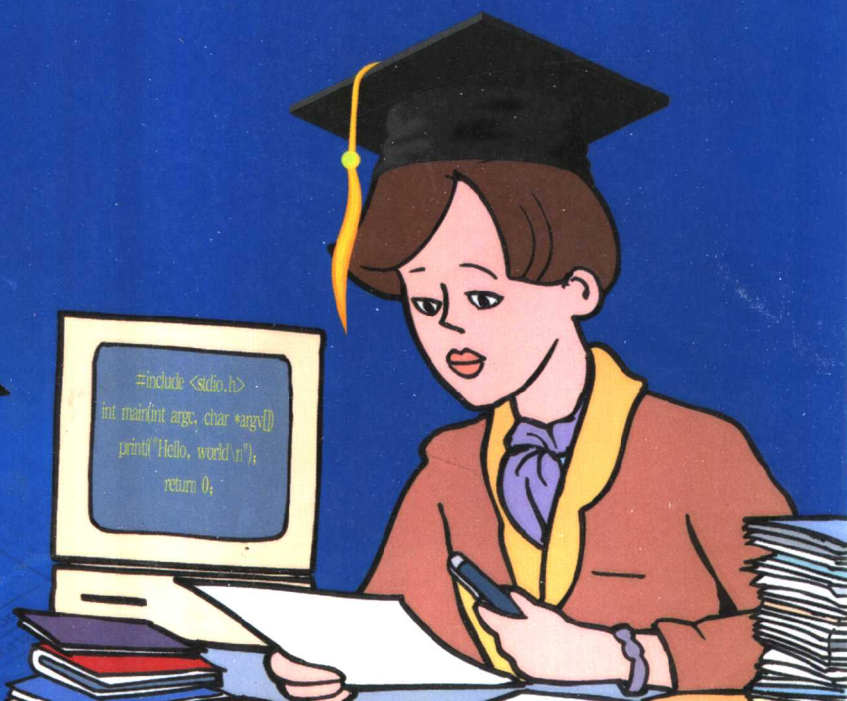
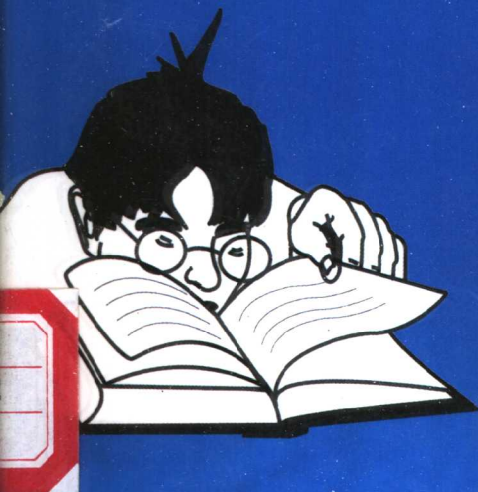
从数学题解

到程序设计



薛维明 编著

薛维明 编著

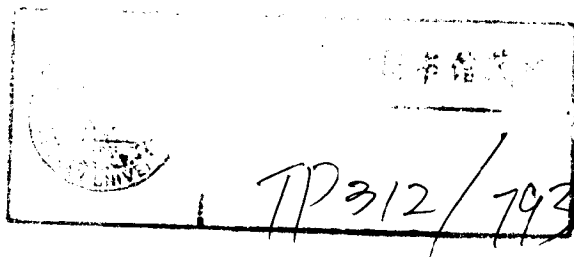


清华大学出版社
<http://www.tup.tsinghua.edu.cn>

► 中学信息技术创新丛书

从数学题解 到程序设计

薛维明 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



0378969

-78

05
10
52

(京)新登字 158 号

内 容 提 要

本书根据中学生的认知能力和心理特点,选取了 24 组引人入胜的著名数学问题。通过介绍这些问题的背景和发展过程、问题的经典解法,来讲解如何设计这些问题的计算机算法,并运用 True BASIC 和 Turbo Pascal 语言来实现算法,编写程序进行解题。

本书在介绍问题解法时主要考虑到可阅读性,把古今方法结合起来,使读者理解各种解法的奇妙和卓越之处。而在介绍问题求解的算法设计和程序编制中,主要考虑到大部分学生的可接受性、算法正确性和程序的可阅读性,在优化方面就不作过多的要求。

本书内容丰富,示例生动,深入浅出,难度适中,富有趣味性和启发性,可供中小学作为计算机选修教材和程序设计竞赛辅导资料。

版权所有,盗版必究。

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。

书 名:从数学题解到程序设计

作 者:薛维明

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京市耀华印刷有限公司

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:15 字数:365 千字

版 次:2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

印 数:0001~5000

书 号:ISBN 7-302-05387-1/TP·3166

定 价:18.00 元

前 言

有位数学家说过：“数学就是问题解决的艺术。”实际上，社会要想得到持续的发展，就必须有人不断地提出大量的、重要的、尚未解决（但原则上能解决）的问题。否则，其生命的源泉便会枯竭。因此，敢于和善于提出问题是人类创造力的重要源泉。

科学技术、社会生产和生活中的进步无一不是由于采用了精巧、奇妙、实用的方法使问题得到解决的结果。数学中用一个专有名词来表达对某个问题的解决——那就是“定理”。在计算机信息科学中，某个问题的解决指的是找到了问题的“算法”。有位大师说过：“计算机科学的核心问题是研究和学习解决问题的算法的艺术”。由此可知，提出问题，找到方法，解决问题的能力是素质教育中创新精神培养的关键。

本书采用以小见大的方法，根据中学生的认知能力和心理特点，选取了各种引人入胜的数学名题，这些名题绝大多数是教学代代相传，人们津津乐道的经典初等数学题，其中不少名题曾经启迪过许多数学大师和名家在中学阶段学习数学的兴趣，使他们日后在科学领域大有作为。书中尽可能地做到科学性和趣味性结合，通过介绍这些问题是怎样提出的，怎样解决的，以及解决这些问题的算法和计算机程序实例，让读者在数学之林中漫步，从问题和解决问题中悟出创新之道。不断地从古今人类的智慧中汲取养分，提高自己的创新能力。

在信息技术教育普及的今天，学习和掌握计算机基础知识和基本操作技能，已成为中学生的基本学习任务的组成部分，本书中的程序设计内容是为了让读者借助解决这些数学经典名题的过程，学习并领会程序设计的基本思想和理念，逐步掌握如何建立解题模型，如何进行算法设计，从而培养良好的编程习惯，提高逻辑思维能力。

本书内容是笔者多年来在教学实践过程中接触和积累的素材资料，其中多数是中学教学大纲范围内可涉及的问题。

本书在写作过程中，得到了吴洪来教授和科海夏非彼主编的大力帮助和支持，他们提出了很多宝贵的建议，吴教授在许多关键问题上作了修改。江苏省常熟中学图书馆为本书写作提供了大量国内外资料，周顾萍女士进行了资料的整理和文字的录入，祝宏教老师为本书做了部分插图，在此一并表示致谢。

由于水平有限，书中一定会有错误或不当之处，恳请广大专家和读者不吝指正。

目 录

第 1 章	算法与编程导入.....	1
1.1	算法的概念.....	1
1.2	算法描述.....	3
1.3	程序设计的基本步骤.....	4
1.4	程序的三种基本结构.....	6
1.5	从一个简单例题导入.....	9
第 2 章	勾股定理 勾股数 星际符号.....	11
第 3 章	三角形面积 海伦-秦九韶公式 海伦三角形.....	18
3.1	利用一般公式求三角形面积.....	19
3.2	利用海伦公式求三角形面积.....	20
3.3	秦九韶公式.....	22
3.4	海伦三角形.....	24
第 4 章	象棋 麦子 世界末日.....	28
4.1	象棋与麦子.....	28
4.1.1	先求幂然后求和.....	29
4.1.2	用等比数列求和.....	31
4.1.3	利用等比数列求和公式直接计算麦子数.....	32
4.2	世界末日问题.....	32
第 5 章	素数 费马数 哥德巴赫猜想.....	38
5.1	素数.....	38
5.2	费马数.....	39
5.3	素数的判定.....	49
5.4	哥德巴赫猜想.....	50
第 6 章	因子分解 现代密码技术.....	56
第 7 章	完全数 相亲数.....	61
7.1	完全数.....	61
7.2	相亲数.....	64



第 8 章 梅桑数 梅桑素数.....	69
第 9 章 欧几里德 最大公约数 最小公倍数.....	73
9.1 最大公约数.....	73
9.2 最小公倍数.....	77
第 10 章 水仙花数 玫瑰花数 五角星数.....	81
10.1 水仙花数.....	81
10.1.1 方法 1.....	81
10.1.2 方法 2.....	84
10.2 玫瑰花数.....	85
10.3 五角星数.....	87
第 11 章 贾宪 杨辉 帕斯卡.....	89
11.1 杨辉三角形.....	89
11.2 帕斯卡三角形.....	92
11.3 用递推法计算杨辉三角形.....	93
11.4 用 11 的幂计算杨辉三角形.....	94
第 12 章 兔子 树枝 斐波那契数列.....	97
第 13 章 鸡兔同笼 百鸡问题.....	104
13.1 鸡兔同笼.....	104
13.2 二元一次方程组.....	106
13.3 三元一次方程组.....	107
13.4 百鸡问题.....	111
13.5 五家共井.....	114
第 14 章 韩信点兵.....	117
第 15 章 墓志铭 不定方程 丢番都问题.....	121
第 16 章 莲花 蜜蜂 古印度方程.....	126
16.1 莲花.....	126
16.2 蜜蜂.....	128
第 17 章 八卦 二进制数.....	132
17.1 八卦与二进制.....	132
17.2 把十进制整数转化为二进制整数.....	133





17.3	把二进制整数转化为十进制整数	135
17.4	把任意十进制数转化为二进制数	137
第 18 章	无理数 $\sqrt{2}e$	139
18.1	计算无理数 $\sqrt{2}$	140
18.2	求 $\sqrt{a}$ 的近似值	144
18.3	" π " 和 " e "	145
第 19 章	谷角猜想	150
第 20 章	尼可马克问题	157
第 21 章	古率 密率 逐步逼近	163
21.1	利用正多边形求 π	164
21.2	利用密率公式计算 π	167
21.3	蒙特卡洛法	170
21.4	级数逼近法	173
第 22 章	河图 洛书 幻方	178
第 23 章	硬币 生日 概率模拟	184
23.1	硬币问题	184
23.2	生日问题	187
第 24 章	故事 模型 解算	189
故事 1	年龄的难题	189
故事 2	年龄的不太难的题	190
故事 3	帽子是谁拿走的?	191
故事 4	一个古怪老板的利润分配方案	192
附录 A	True BASIC 程序设计语言简介	195
A.1	True BASIC 语言基础知识	195
A.1.1	True BASIC 语言	195
A.1.2	True BASIC 语言中的常量、变量、标准函数和表达式	196
A.2	True BASIC 语言的主要语句	199
A.2.1	print 语句	199
A.2.2	let 语句	201
A.2.3	input 语句	202





A.2.4	read/data 语句.....	203
A.2.5	if 型结构控制语句.....	204
A.2.6	for-next 循环语句.....	207
A.2.7	do 循环.....	209
A.2.8	数组.....	213
A.2.9	程序的保存与调用.....	216
附录 B	Turbo Pascal 程序设计语言简介.....	217
B.1	Turbo Pascal 语言基础知识.....	217
B.1.1	Turbo Pascal 程序的结构.....	217
B.1.2	关键词和标识符.....	218
B.1.3	常量和变量.....	219
B.1.4	标准函数和自定义函数.....	220
B.1.5	数据类型.....	221
B.2	顺序结构的程序设计.....	223
B.3	分支结构的程序设计.....	223
B.4	循环结构的程序设计.....	224
B.5	数组.....	224
参考书目		225



第 1 章



算法与编程导入

在我们即将跨入新世纪的时候，国家教育部制订了《中小学信息技术课程指导纲要（试行）》，其中已将“程序设计”作为高中教学内容的基本模块，教学目标明确要求学生“了解程序设计的基本思想，培养逻辑思维能力。”

中国工程院院士计算机科学家李国杰说道：“事实上，电脑发展到今天，能有如此广泛而神奇的应用，除了半导体集成电路芯片制造工艺提高以外，主要靠软件，而软件的核心是算法（不是编程技巧）。……算法设计是人类智慧的结晶，计算机科学中的知识创新主要就是算法的创新，创建一种新算法其意义不亚于建造一种新机型。……严格来说，不讲算法，计算机科学就无从说起。”

学习计算机程序设计，对培养学生学习信息技术的兴趣，掌握信息技术的基本知识和技能，以及形成良好文化素养，将会起到举足轻重的作用。信息技术课程是中学一门知识性与技能性相结合的基础工具课程。学习计算机程序设计，对培养学生的逻辑思维能力、创新精神和解决实际问题的能力大有好处。

本书选取了一系列引人入胜的著名的初等数学问题。这些问题的本身及它们的解法，曾在国内外著名数学家成长的道路上起过重要作用，直到晚年有的科学家还在自己的传记或著作中津津乐道。在中学阶段，这些数学名题在他们成长道路上所起的作用是不可磨灭的。

我们在书中带领广大同学漫步于数学之林。在介绍问题的经典解法的同时，特别注重探讨如何设计这些问题的计算机算法，并运用 True BASIC 和 Turbo Pascal 语言来实现算法，编写程序进行解题。书中我们努力针对中学生的特点和认知能力，启发他们的创新意识，培养他们的创新能力，发挥他们的聪明才智。

1.1 算法的概念

早在公元 825 年左右，一位名叫阿尔·花拉子米（Al Khowârizmî）的波斯数学家写了一本书，书中概括了进行数学四则运算的法则。现代名词“算法”



(algorithm) 就来源于这位数字家的名字。

所谓算法就是逐步执行某类计算的方法，所谓找到了某种算法是指使用有精确定义的一系列运算规则能在有限步骤内求解某种问题，其中的每条规则必须是精确的，能行的，没有二义性的，同时应对所有的初始数据，而不仅是对某些特殊数值有效。算法是从一个初始步骤开始，每一个步骤只能有一个确定的后继步骤，从而组成一个步骤序列，这个序列必须是有限的，序列的终止表示问题得到解答或指出问题没有解答。

算法的问题实际上是只有空间和时间两个方面，即为实现算法要花费多大的机器存储空间和多长的计算时间，当然这两方面的问题在一定条件下可相互转化，懂得编程之道的人，总会有足够的时间和空间来达到他们的目标。

初学者往往认为计算机解题十分神秘，实际上，计算机解题的过程与我们处理事情十分相似。人们做任何一件事都有一个过程，这个过程就是在规定的条件下逐步执行操作步骤的顺序。例如，上化学实验课时，稀释浓硫酸的操作序列是：取容器→加水→注入适量浓度的硫酸→轻轻加以搅拌。这个步骤序列是不能违反的。同样，烧饭、做菜、机床加工、生产计划的制订等等，都有基本操作序列，只不过在不相同的环境和条件下对操作步骤和种类有不同的规定罢了。一般地说，这种操作序列也是一种“算法”。下面列举三个数学中的例子。

【例 1】两个变量值的交换。

已知变量 x 和 y 分别存放了数据，现在要交换这两个变量值。为了达到交换的目的，需要引进一个中间变量 s 。其算法如下：

- (1) $s \leftarrow x$ (首先将 x 的值赋给变量 s ，这时变量 x 的空间就腾出来了)
- (2) $x \leftarrow y$ (接着将 y 的值赋给变量 x ，这时变量 y 的空间就可作其他用途了)
- (3) $y \leftarrow s$ (最后将 s 的值赋给变量 y ，就完成了 x 和 y 两个变量值交换的任务了)

经过上述三个步骤可以完成交换任务。两个变量值的交换在程序设计中经常要用到，必须掌握这一基本方法。这个方法的记忆特点是：头尾一样，中间两两一样。试想一下，如果不引入变量 s 作中间过渡，我们还能完成这一任务吗？





【例 2】输入三个不相同的数，求出其中的最小数。

设已输入了 a, b, c 三个不相同的数后，我们设置一个备用变量 $\min$ ，用于存放最小数，先将 a 与 b 进行比较，把小者送给变量 $\min$ ，再把 c 与 $\min$ 进行比较，若 $c < \min$ ，则将 c 的数据送给变量 $\min$ ，最后变量 $\min$ 中的数据就是 a, b, c 三个数中的最小数。

现将具体算法叙述如下：

- (1) 对任意给定的变量 a, b, c ， $\min$ 为一临时变量。
- (2) 若 $a < b$ ，则 $\min \leftarrow a$ ，否则 $\min \leftarrow b$ 。
- (3) 再将 c 与 $\min$ 进行比较，若 $c < \min$ ，则 $\min \leftarrow c$ 。
- (4) $\min$ 中的数值即为问题的解。

【例 3】计算自然数列前 100 项之和。

从 1 开始，依次将自然数 1, 2, 3, 4, 5, ..., 98, 99, 100 依次相加，加到 100 为止。其算法可以写成如下形式：

- (1) 设 $\text{sum} = 0$ 。
- (2) 设 $i = 1$ 。
- (3) $\text{sum} \leftarrow \text{sum} + i$ 。
- (4) 使 i 值增加 1，即 $i \leftarrow i + 1$ 。
- (5) 若 $i \leq 100$ ，则返回步骤 (3)，否则算法停止。

最后 sum 的数值即是自然数中前 100 个自然数之和。

1.2 算法描述

算法是解题步骤精确的描述，可以用自然语言、流程图或程序语言方式进行描述。

用自然语言描述算法，人们比较容易接受，但是叙述较繁琐冗长，且易出现“二义性”，故一般较少采用这种方法。采用流程图描述算法，能够将算法清晰、直观地表示出来，所以本书采用流程图的方式来描述算法。





1.3 程序设计的基本步骤

为了使初学者能形成良好的程序设计习惯，设计出质量较好的程序，下面给出程序设计通常所用的 5 个基本步骤：

1. 需求分析

要求计算机去解决实际问题，首先要进行审题，即要认真分析题目，需要解决什么问题，达到什么目标，然后从已知条件（注意题目的隐含条件）出发，分析经过哪些处理才能够解决问题。只有这样正确地分析需求，才能更好地为设计算法、画出流程图和编写程序打好基础，否则就会“差之毫厘，失之千里”。

2. 算法设计

算法设计的重点是研究如何对一个给定的问题设计出解决它的有效算法以及探讨算法设计的一般技巧，是理论计算机科学的一个重要分支，它不仅是指计算的方法，而且还包含从何处着手、解题步骤以及结果处理等全过程。例如，邮寄一封信的过程可分为写信、写信封、贴邮票、投入信箱等 4 个步骤，这些处理可以看作是写信的算法。当然这是很简单的算法，利用计算机解决问题时所采用的算法往往要复杂得多。计算机科学家已创建出许多行之有效的通用算法，而且还在日新月异地发展着。可以说，算法设计是程序设计过程中，也是整个解题过程中最关键的一步。

3. 画流程图

算法设计好了，为了将它比较清晰、直观地描述出来，常常需要将设计方案表示出来，画流程图是这种表示的方法之一。邮寄信件算法的流程图如图 1 所示。



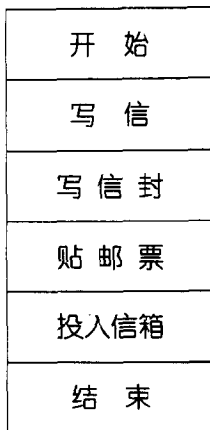


图 1 寄信的流程图

可见，流程图确实能够较好地反映解决问题的操作过程。因此，画流程图是应用计算机解决实际问题的一个行之有效的方法。

4. 编写代码

用程序语言来实现算法代码化，通俗地说就是编写程序。经过以上三个步骤，就可以着手编写程序了。编写程序就是用计算机语言（机器语言、汇编语言、高级语言）来描述或实现算法，这是告诉计算机如何一步步地去解决问题的一系列命令或语句。本书根据目前多数中学实际教学情况，采用了 True BASIC 和 Turbo Pascal 两种程序设计语言编写程序。

5. 上机调试

程序编好以后，应该进行严格地测试，在调试过程中发现和纠正错误，直至程序能正常运行，如果不经调试，就直接让它运行，往往会发现许多错误。因此，要反复排错和纠正，直到程序正确通过，运行出正确结果。所以说，上机调试是一项十分重要的工作。





1.4 程序的三种基本结构

为了进一步提高程序设计的质量，必须学习结构化程序设计方法。结构化程序设计是一种重要的程序设计方法，这种方法最初由 IBM 公司史蒂文斯(Stevens)、迈尔斯(Myers)等人提出。其基本思想是将系统设计成由相对独立的、单一功能的模块组成的结构，在模块之间按层次结构组织起来，这也是一种化大(模块)为小(模块)，化难为易的办法，遵照这种方法所产生的程序具有流程清晰，可阅读性、可维护性良好等特点。在程序组织的逻辑结构化方面，拜姆(Böhm)和雅可皮尼(Jacopini)提出了一个结构化定理，证明了用三种基本结构就可以表示任何具有一个入口和一个出口的程序逻辑，这三种结构就是：顺序结构、分支(选择)结构和循环(重复)结构。事实上，任何计算机程序都可以由这三种结构组合而成。

1. 顺序结构

顺序结构是一种简单的基本结构，如图 2 所示，计算机依次进行多个处理或执行多条指令语句 $A_1, A_2, A_3, \dots, A_n$ 。例如，两个变量值的交换，我们采用如图 3 所示顺序结构流程图。

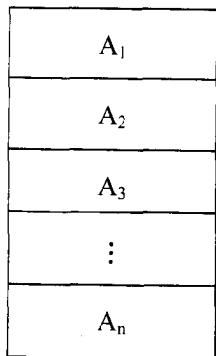


图 2

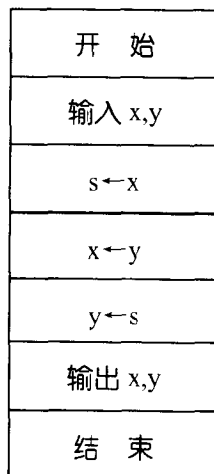


图 3





2. 分支（选择）结构

在编写程序过程中，仅有顺序结构显然是不够的，有时需要根据某个条件进行判断，再决定选择流程走向，即下一步做什么，这就要使用分支结构，如图 4 所示。例如，输入三个不相同的数，求出其中的最小数，就可以采用如图 5 所示的分支结构流程图。

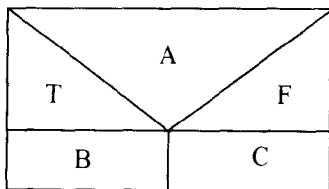


图 4

图 4 的意思是，若 A $\begin{cases} \text{为真(T)，继续执行 B} \\ \text{为假(F)，继续执行 C} \end{cases}$

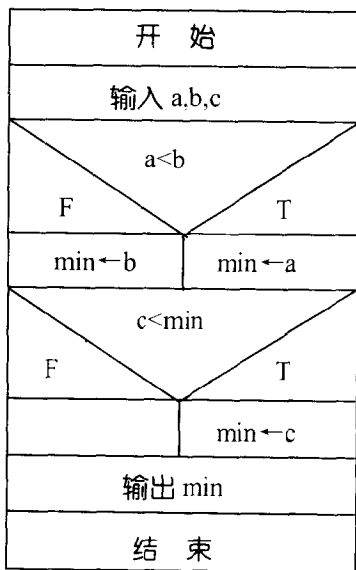


图 5





3. 循环（重复）结构

在编写程序的过程中，经常需要重复执行同一串操作，我们称它为循环结构。循环结构常用的有两种类型：当型循环结构（见图 6）和直到型循环结构（见图 7）。

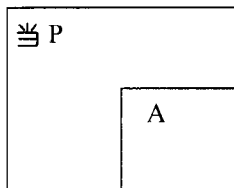


图 6

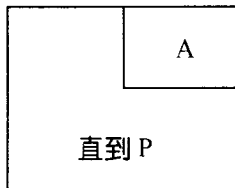


图 7

进入当型循环后，若条件 P 不成立，则不执行循环体 A ；若条件 P 成立，则执行循环体。正确的必须在循环体执行有限次后，条件 P 不再成立，从而退出循环。

直到型循环结构的循环体至少被执行一次，通常直到型循环体执行有限次后，条件 P 成立，就退出循环。例如，求自然数列前 100 项之和，就可以采用如图 8 所示的流程图表示直到型循环结构。

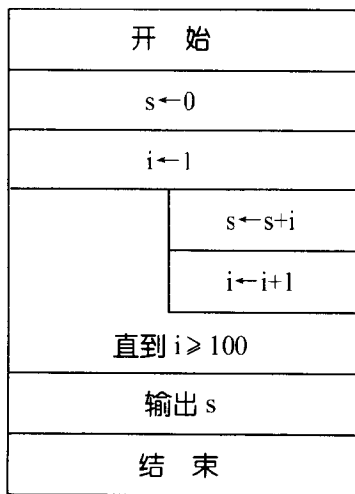


图 8





1.5 从一个简单例题导入

为了更好地学习后面的内容，首先让我们来看一个简单的例题。

【例 4】已知圆的半径为 10cm，求圆的面积。

1. 需求分析

已知圆的半径，要求计算此圆的面积值。

2. 算法设计

半径为 r 的圆面积计算公式为：

$$S = \pi r^2$$

本例中圆半径为 10cm，计算圆面积的算法如下：

- | | |
|----------------------------|--------------------|
| (1) $r \leftarrow 10$ | (把半径 10 赋给变量 r) |
| (2) $s \leftarrow \pi r^2$ | (利用上面的公式计算圆的面积) |
| (3) print "s=";s | (输出圆的面积 s) |
| (4) end | (结束) |

3. 画流程图

