工业和信息化“十三五”
人才培养规划教材

计算机数学

算法基础 线性代数与图论

Computer Mathematics

邓洁 桂改花 ◎ 主编
康海刚 ◎ 副主编

“**数学理论知识** + **专业技术应用**”的编写方向
紧贴计算机相关专业对**数学知识**、**思维训练**的需要
让读者真实地认识到**数学方法**和**模型**对计算机技术的**重要性**

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化“十三五”
人才培养规划教材

计算机数学

算法基础 线性代数与图论

Computer Mathematics

邓洁 桂改花 ◎ 主编
康海刚 ◎ 副主编

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

计算机数学 ：算法基础 线性代数与图论 / 邓洁，桂改花主编. -- 北京 ：人民邮电出版社，2016.8
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-42638-3

Ⅰ. ①计… Ⅱ. ①邓… ②桂… Ⅲ. ①电子计算机－数学基础－高等学校－教材 Ⅳ. ①TP301.6

中国版本图书馆CIP数据核字(2016)第162288号

内 容 提 要

本书针对计算机相关专业对数学课程的需求编写而成，共分为 6 章，详细讲述了包括算法基础、向量与矩阵、图形变换的矩阵方法、线性方程组、图与网络分析、树、MATLAB 入门等内容。

本书在内容的选取上遵循“应用导向，必需够用”的原则，以计算机图形变换实现、Google 网站排名算法、网络分析中的最短路算法、最小连接算法、数据挖掘中的决策树算法等为应用背景，重点介绍了工科学科中不可缺少的数学工具——向量、矩阵和线性方程组，充分体现了为计算机相关专业服务的理念。

本书可作为高等院校计算机相关专业的数学教材，也可供工科技术人员参考。

◆ 主　　编　邓　洁　桂改花
副 主 编　康海刚
责任编辑　范博涛
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：12.5　　2016 年 8 月第 1 版
字数：309 千字　　2016 年 8 月北京第 1 次印刷

定价：29.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315

前言

数学课程作为高等教育的一门公共基础课，其基础性、工具性的地位被广泛认可，尤其在训练思维、提供方法、建立模型等方面有着无可替代的作用。

计算机数学作为计算机相关专业的一门专业基础课，不仅为其他课程的学习提供必要的知识和思维训练，也为培育学生的科学素养提供了重要的途径。实践表明，数学思维的训练，尤其是数学建模的训练不仅对学生的日常专业学习有着很大的帮助，还对学生获取专业技能大赛的奖项有着巨大贡献。

目前高等院校计算机相关专业学生的来源广泛，他们的数学基础千差万别。新生入学常常会问，学数学有什么用？专业上用得到吗？这些疑问长期困扰着学生，也导致教师授课辛苦，却没有成就感。

为改变这种困境，我们总结了国内外众多计算机数学教材的编写思路，确定“数学理论知识+专业技术应用”的编写方向，紧贴计算机相关专业对数学知识、思维训练的需要，把数学课程作为学生学习数据结构、程序设计、数据库等专业课程的前导课程。通过我们的努力，学生不仅能掌握数学的基础知识，还能真实地认识到数学的方法和模型对计算机技术的重要性。例如，矩阵和线性方程组在 Google 网站排名算法中的运用，图论模型在网络分析中的运用，以及当前热门的数据挖掘技术中决策树算法的运用。此外，本书在附录中还介绍了工程软件 MATLAB 的基本操作，供需要的读者学习。

本书由广东科学技术职业学院的邓洁、桂改花任主编，康海刚任副主编。

由于编者水平有限，书中难免有错误，敬请广大读者批评指正！

编者

2016 年 5 月

目 录 CONTENTS

第六章 树 137

附录 A MATLAB 入门 158

参考文献 191

PART 1

第一章 算法基础

本章介绍算法的含义、算法的基本逻辑结构、递归算法及其实例。
1.1 节介绍算法的含义、算法的特性、算法的表示。
1.2 节介绍算法的三种逻辑结构，能分析简单问题的算法并用图描述。
1.3 节介绍递归算法的思想，了解递归逻辑过程，掌握求最大公约数的递归方法并能编写算法。

电子计算机自发明并于 1946 年 2 月 15 日在美国宾夕法尼亚大学正式投入使用以来，更新换代非常迅速，现代计算机系统的功能越来越强大，应用领域越来越深入、广泛，计算机、手机已成为人们日常活动中必不可少的工具。我们知道，计算机解决任何问题都是靠程序驱动完成的。指挥计算机进行操作的一连串指令序列称为程序。计算机的基本原理是存储程序和程序控制，计算机程序可描述为程序=算法+数据。算法是什么呢？简单说，算法=逻辑+控制。计算机技术发展日新月异，但基本功能与原理并没有发生变化，其最基本的功能是执行二进制数算术运算和逻辑运算。本章将学习有关算法的基础知识。

为计算机发明奠基的数学家

主宰世界的 10 大算法

推荐阅读链接：

1.《为计算机发明奠基的数学家》

2.《主宰世界的 10 大算法》

1.1 算法

1.1.1 什么是算法

算法（Algorism）一词最初出现在 12 世纪，是用于表示十进制算术运算的规则。18 世纪，算法 Algorism 演变为 Algorithm，算法概念有了更广的含义。任何定义明确的计算步骤都可称为算法，或者说算法是合乎逻辑、简捷的一系列步骤。现在算法通常指可以用计算机来解决某一类问题的程序或步骤。

1.1.2 算法的特性

问题不同，解决的思路和采取的方法与步骤就有针对性，所以对应的算法也各不相同。

但各种算法有如下共同之处：首先计算机要有操作对象，通过输入，给予计算机问题所涉及的对象；最后要能得到运行结果，即有输出；在输入与输出之间是具体的方法和步骤，这些方法和步骤必须是确定的、正确的、有限的、有效的、通用的。因而，运行于计算机的各种算法有如下特征。

（1）输入：算法从一个指定集合得到输入值，可以有 0 个、1 个或多个值，由赋值或输入语句实现；

（2）输出：对每个输入值，算法都要从指定的集合中产生输出值，输出值就是问题的解，可以有 1 个或多个输出值，由输出语句实现；

（3）确定性：算法的步骤必须准确定义，不能产生歧义；

（4）正确性：对每一次输入值，算法都应产生正确的输出值；

（5）有限性：对任何输入，算法都应在有限步骤之后产生输出；

（6）有效性：算法每一步必须能够准确地执行，并在有限时间内完成；

（7）通用性：算法不只是用于特定的输入值，应该可以用于满足条件的所有问题。

例 1.1 找出计算机软件专业录取的新生中高考总分的最高分。

分析：这个问题等价于求有限整数序列中最大值的算法，可采取以下步骤。

（1）将序列中第一个整数设为临时最大值（max）；

（2）将序列中下一个整数与临时最大值比较，如果这个数大于临时最大值，临时最大值更新为这个整数；

（3）重复第（2）步，一直比较到序列中最后一个数时停止。此时临时最大值就是序列中的最大整数。

在此算法中，输入是软件专业所有新生的高考成绩，输出是高考最高分，算法过程从序列第一项开始，并把序列第一项设为临时最大值的初始值，接着逐项检查，如果有一项超过最大值，就把最大值更新为这一项的值，检查到序列的最后一项结束。算法每进行一步，要么是比较最大值和这项的大小，要么是更新最大值的值，所以每一步的操作都是确定的，能保证最大值是已检查过的最大整数，结果是正确的。如果序列包含 n 个整数，经过 n-1 次比较就结束，所以算法步骤是有限的、有效的。这个算法可以用于求任何有限整数序列问题的最大元素，所以它是通用的。

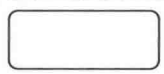
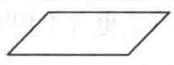
1.1.3 算法的表示

算法可以用自然语言、程序框图、N-S 图、伪代码、计算机语言表示。例 1.1 就是用自然语言描述求整数序列最大值的算法。

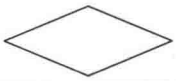
1. 程序框图

程序框图又叫流程图，是由一些规定的图形、流程线和文字说明来直观描述算法的图形。程序框及其说明如表 1.1 所示。

表 1.1 程序框及其说明

程序框	名称	功能
	起止框	表示一个算法的起始和结束
	输入、输出框	表示一个算法的输入和输出的信息

续表

程序框	名称	功能
	执行框	赋值、计算
	判断框	判断某一条件是否成立，成立时在出口处标明“是”或“Y”；不成立时标明“否”或“N”

例 1.2 画出例 1.1 的算法的流程图（见图 1.1）。

2. N-S 图

流程图由一些特定意义的图形、流程线及简要的文字说明构成，它能清晰、明确地表示程序的运行过程。因为在使用过程中发现流程线不是必需的，人们设计了一种新的流程图，它把整个程序写在一个大框内，这个大框图由若干个小的基本框图构成，这种流程图简称 N-S 图。N-S 图是无线的流程图，又称盒图，在 1973 年由美国两位学者 I.Nassi 和 B.Shneiderman 提出。

例 1.3 例 1.1 算法的 N-S 图（见图 1.2）。

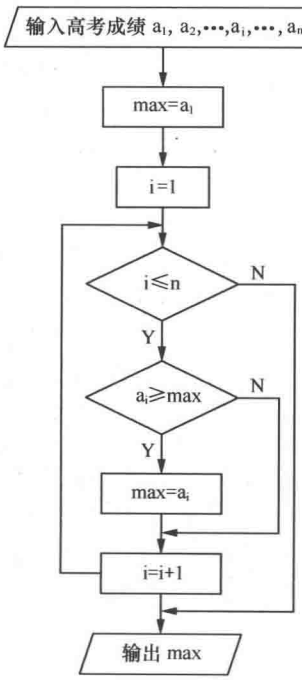


图 1.1

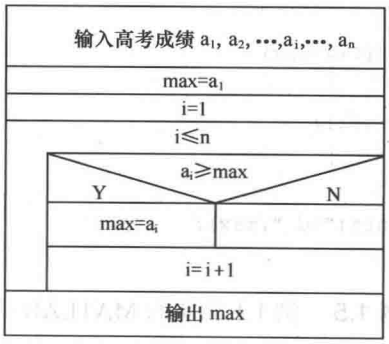


图 1.2

3. 伪代码（Pseudocode）

伪代码是一种介于自然语言与编程语言之间的算法描述语言，便于理解，并不依赖于语言，它用来表示程序执行过程，而不一定能编译运行的代码。使用伪代码的目的是为了使被描述的算法可以容易地以任何一种编程语言实现。

例如：

IF 九点以前 THEN

做私人事务；

ELSE 9 点到 18 点 THEN
    工作;
    ELSE
    下班;
    END

4. 计算机语言（Computer Language）

计算机语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言三大类。

计算机所能识别的语言只有机器语言，即由 0 和 1 构成的代码。但通常人们编程时，并不采用机器语言，因为它非常难于记忆和识别。汇编语言的实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的标识符，更容易识别和记忆。高级语言是目前绝大多数编程者的选择，它并不是特指某一种具体的语言，而是包括了很多编程语言，如目前流行的 C、C++、C#、Java、VB、VC、FoxPro、Delphi 等，这些语言的语法、命令格式都各不相同。

例 1.4 例 1.1 算法的 C 语言程序。

```
#include <stdio.h>
int main()
{
int a[100],i,n,max;
scanf("%d",&n);
for(i=0;i<n;i++)
    {
scanf("%d",&a[i]);
    }
max=a[0];
for(i=1;i<=n-1;i++)
    {
if(a[i]>=max)
        {
max=a[i];
        }
    }
printf("%d ",max);
}
```

例 1.5 例 1.1 算法的 MATLAB 语言程序。

```
x=input('x=');
n=length(x);
max=x(1);
for i=1:n
if x(i)>=max
max=x(i);
end
i=i+1;
end
fprintf('max=%d\n',max)
```

1.2 算法的逻辑结构

1.2.1 算法的基本逻辑结构

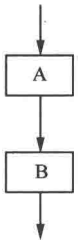
算法控制着各条指令的运行次序，规定语句的逻辑结构。算法包含三种基本逻辑结构：顺序结构、条件结构和循环结构。任何由计算机程序处理的问题都可以表示为基本结构或基本结构的组合。

1. 顺序结构

顺序结构是指按顺序执行完一步后再执行下一步的执行结构。顺序结构在程序框图中的体现是用流程线将程序框自上而下地连接起来，并按顺序执行算法步骤，如图 1.3 所示。

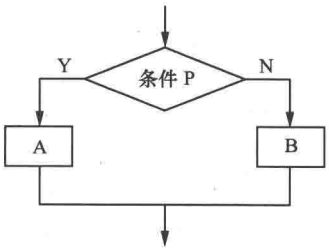
2. 条件结构

条件结构（也称选择结构、分支结构）在程序框图中用判断框来表示，判断框内写条件，两个出口分别对应着条件满足和条件不满足时所执行的不同指令，如图 1.4 所示。



顺序结构

图 1.3

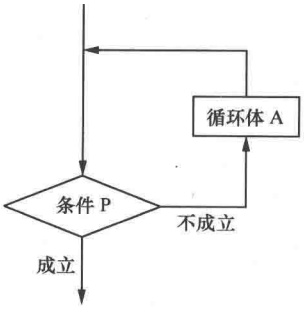


条件结构

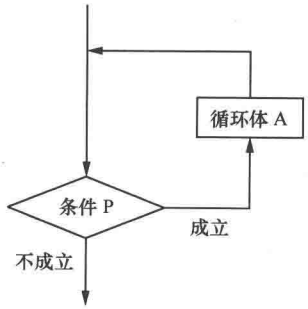
图 1.4

3. 循环结构

在一些算法中，会出现从某处开始，按照一定条件，反复执行某一步骤，这就是循环结构。反复执行的步骤称为循环体。循环结构有三个要素：循环变量、循环体和循环终止条件。循环结构必然包含条件结构，循环结构在程序框图中利用判断框来表示，判断框内写条件，两个出口分别对应着条件成立和条件不成立时所执行的不同指令，其中一个要指向循环体，然后再从循环体回到判断框的入口处。循环结构有两种类型：当型和直到型。当型结构指当条件满足时，反复执行循环体，不满足则停止。直到型结构指在执行了一次循环体之后，对控制循环条件进行判断，当条件不满足时执行循环体，满足为止。常按照“确定循环体→初始化变量→设定循环终止条件”的顺序来构造循环结构，如图 1.5 所示。



直到型循环结构



当型循环结构

图 1.5

4. 三种基本逻辑结构的 N-S 图

三种基本逻辑结构的 N-S 图如图 1.6 所示。

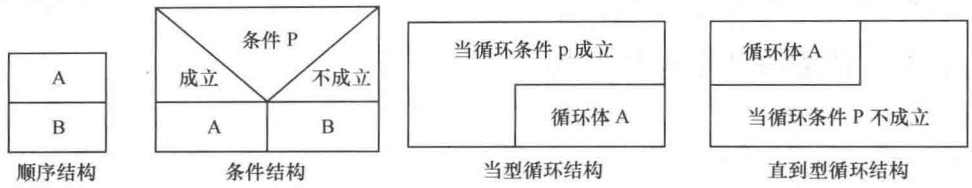


图 1.6

1.2.2 算法举例

例 1.6 设计一个求解一元二次方程 $ax^2+bx+c=0$ 的算法。

算法分析：我们知道，根据判别式 $\Delta=b^2-4ac$ 的符号，一元二次方程 $ax^2+bx+c=0$ 的解有三种情况，所以，在求解方程前要先判断判别式的符号，然后执行不同的计算。

第一步：输入三个系数 a、b、c；

第二步：计算 $\Delta=b^2-4ac$ 的值；

第三步：判断是否成立 $\Delta\geqslant 0$，若是，则计算 $p=-\dfrac{b}{2a}$，$q=\dfrac{\sqrt{\Delta}}{2a}$，进入第四步；若否，输出“方程没有实数根”。

第四步：判断 $\Delta=0$，若是，则输出 $x=p$；若否，计算 $x_1=p+q$，$x_2=p-q$，并输出 x_1、x_2，结束。

例 1.6 的 N-S 图如图 1.7（a）所示，流程图如图 1.7（b）所示。

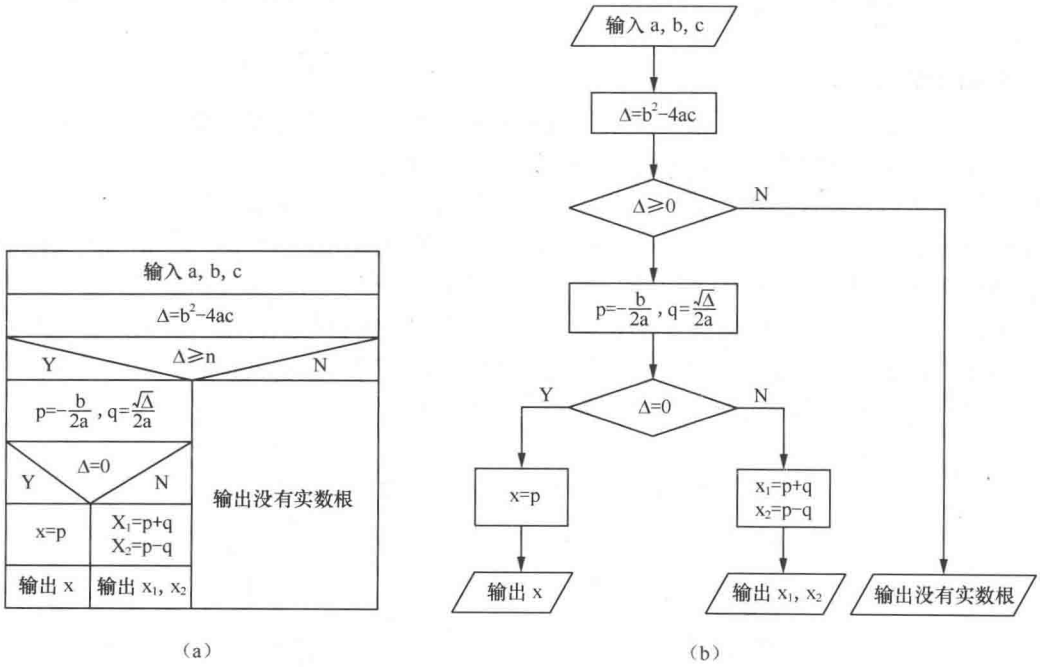


图 1.7

例 1.7 中国农历 60 年一大轮回，按天干（甲、乙、丙、丁、戊、己、庚、辛、壬、癸）和地支（子、丑、寅、卯、辰、巳、午、未、申、酉、成、亥）循环排列而成。天干共 10 个，公元纪年也是 10 年一周期，公元纪年除以 10 的余数（就是公元纪年的尾数）与天干之间有

一种关联，即余数与天干一一对应（见表 1.2）。

表 1.2　余数与天干对应表

余数	4	5	6	7	8	9	0	1	2	3
天干	甲	乙	丙	丁	戊	己	庚	辛	壬	癸

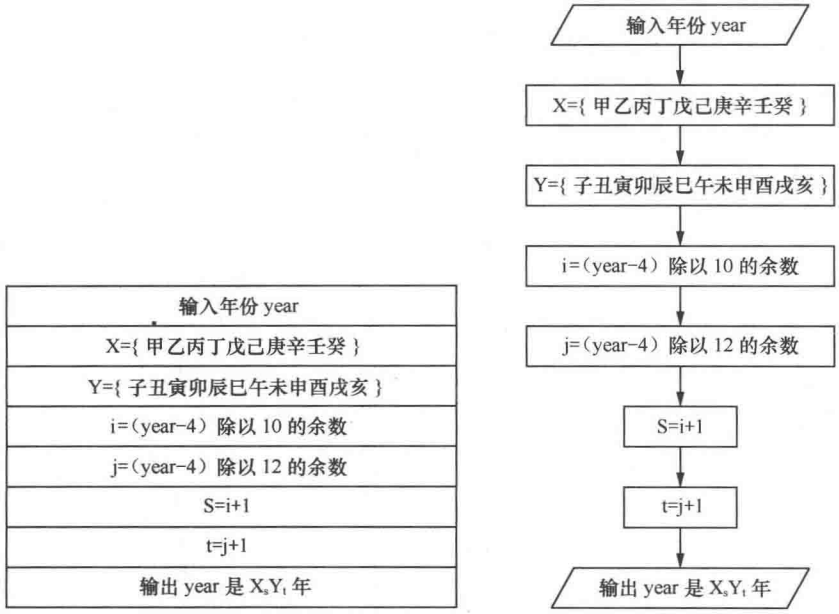
地支共 12 个，地支 12 年一轮回，用公元纪年除以 12，余数与地支也有一一对应关联（如表 1.3 所示）。

表 1.3　余数与地支对应表

余数	4	5	6	7	8	9	0	1	2	3	4	5
地支	子	丑	寅	卯	辰	巳	午	未	申	酉	戌	亥

根据以上对应关系，设计一个算法并输出字符串农历纪年的算法，然后画出 N-S 图。

解：N-S 图如图 1.8（a）所示，流程图如图 1.8（b）所示。



（a）　　（b）

图 1.8

例 1.8　设计一个求 10! 的算法，画出 N-S 图。

算法分析：求 10!，要先计算 1×2，然后将 1×2 的结果乘以 3，前 3 个数的乘积再乘以 4，以此方式进行一个累乘循环的计算过程。所以，需要设定一个存放每次乘积的变量 t 和一个循环计数变量 i，将累乘变量的初值设为 1，计数变量在 $1 \leqslant i \leqslant 10$ 范围。

第一步：赋初值 $t=1$，$i=1$。

第二步：判断 $i \leqslant 10$，若是，计数变量 i 增加 1，即 $i=i+1$，累乘变量 t 乘以计数变量，即 $t=t \times i$；若否，输出 t，结束。

例 1.8 的 N-S 图如图 1.9（a）所示，流程图如图 1.9（b）所示。

例 1.9 用“二分法”设计一个求方程 $x^2-2=0$ 的近似根的算法。

二分法是基于“根的存在定理”，求方程根的近似值的一种算法。二分法思想为：将函数的零点所在区间不断地一分为二，使所得的新区间不断变窄，两个端点逐步逼近零点，达到精度要求为止。

根的存在定理：设函数 $f(x)$ 在闭区间 $[a, b]$ 上连续，且 $f(a)\times f(b)<0$，则（a，b）内至少有一点 c，使得 $f(c)=0$。c 称为函数 $f(x)$ 的零点，这个定理可以帮助我们确定方程根的大致范围，或判断方程在某一范围内是否有解。

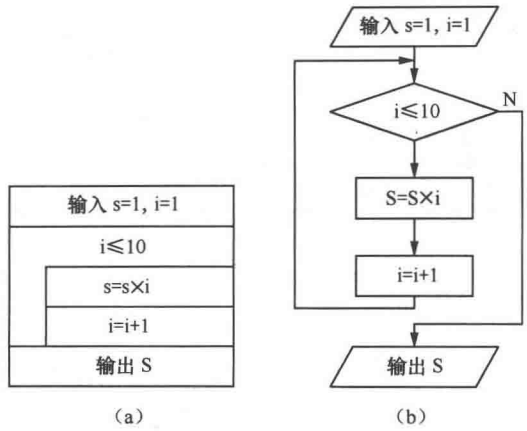


图 1.9

算法分析：根据“二分法”步骤，需要的已知条件有方程、有且只有一个根的区间、近似根的精度。

第一步：输入 $f(x)=x^2-2$，误差精度 ε，a、b（a、b 为有根区间的端点）。

第二步：令 $m=\dfrac{a+b}{2}$，判断 $f(m)$ 是否为 0，若是，则 m 为方程的根；若否，进入第三步。

第三步：判断 $f(a)\times f(m)<0$，若是，则令 $b=m$；若否，令 $a=m$。

第四步：判断 $|a-b|<\varepsilon$，若是，则输出 $x=m$；若否，则返回第二步。

例 1.9 的 N-S 图如图 1.10（a）所示，流程图如图 1.10（b）所示。

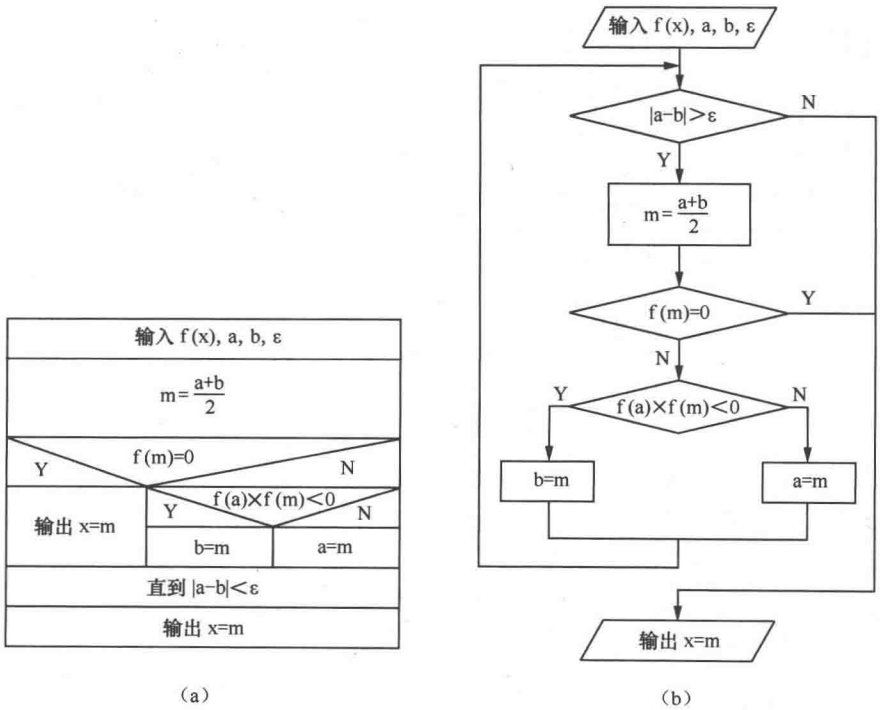


图 1.10

课堂练习 1.2

1. 判断下列说法是否正确。

A 算法的三种基本逻辑结构流程图都只有一个入口、一个出口。（　　）

B 循环结构有选择性和重复性，选择结构具有选择性但不重复。（　　）

2. 填空。

（1）算法的基本逻辑结构有（　　）。

（2）表示算法的图有（　　）。

（3）求 10! 的算法里循环结构的三个要素（　　）。

（4）表达交换 a、b 的值的语句是（　　）。

3. 已知华氏温度和摄氏温度的转换公式是：

$$（华氏温度-32）\times\frac{5}{9}=摄氏温度$$

设计一个将华氏温度转换成摄氏温度的算法，并画出其流程图或 N-S 图。

4. 设计一个算法，求方程 $ax+b=0$ 的根，并画出其流程图或 N-S 图。

5. 设计一个算法，求 $1+2+4+\cdots+2^{49}$，并画出其流程图或 N-S 图。

1.3 递归算法

小游戏：汉诺塔或梵塔问题。

有 3 个基座 A、B、C，开始时 A 基座上有 n 个大小不同的盘子，大盘子在下、小盘子在上叠在一起，问：要把 A 基座上的 n 个盘子移到 C 基座上，每次只能移动一个，并且移动过程中始终保持大盘子在下、小盘子在上，能做到吗？如能做到，要移动几次？

（1）设 n 个大小不同的盘子按规则移动，需要移动 $f(n)$次，请动手做一做，算出以下答案。

$f(1)=$　　$f(2)=$　　$f(3)=$　　$f(4)=$

（2）试猜想 $f(n)=2f(n-1)+1$ 是否成立？ $f(n)=2^n-1$ 是否成立？

证明：$f(n)=2f(n-1)+1 \Rightarrow f(n)+1=2[f(n-1)+1] \Rightarrow \dfrac{f(n)+1}{f(n-1)+1}=2$

所以，$\{f(n)+1\}$ 构成一个等比数列，首项为 $f(1)+1=2$，公比为 2，则

$f(n)+1=2\times 2^{n-1}=2^n$，即 $f(n)=2^n-1$。

（3）解决这个问题能否由计算机实现？过程怎样？——可用递归算法编程实现。

1.3.1 什么是递归

我们知道，用解析法表示函数常有：显函数形式 $y=f(x)$、隐函数形式 $f(x,y)=0$、参数方程形式 $\begin{cases}x=x(t)\\y=y(t)\end{cases}$（$t$ 为参数）。在学习数列时，通项公式可确定一个数列，递推公式也可确定数列。如斐波那契数列（1，1，2，3，5，8，13，21，…）是用 $a_1=1$、$a_2=1$ 以及当 $n>2$ 时，$a_n=a_{n-1}+a_{n-2}$ 定义的数列。函数也有递归定义形式，如 $f(n)=2f(n-1)+1$。计算机术语中，实现某些功能的程序段也称为函数。各种计算机高级语言都有函数的嵌套调用和递归调用，什么是递归呢？

1. 递归

自己调用自己，称为递归。自己调用自己的函数，称为递归函数。递归函数包括直接递归和间接递归。直接递归是指函数 F 的代码中直接包含了调用 F 的语句，而间接递归是指函数 F 调用了函数 G，G 又调用了 H，如此进行下去，直到 F 又被调用。

如求 n 的阶乘运算，定义：$n!=(n-1)!\times n$。定义中 $n!$ 与 $(n-1)!$ 的算法是相同的，本质上它们是同一函数，求 $n!$，先要求 $(n-1)!$，所以，阶乘函数体现了自己调用自己。

例 1.10 由初值 $y_0=1.41$ 和递推关系 $y_n=10y_{n-1}-1$（n=1, 2, 3, 4, …）确定的数列（函数），设 $f(n)=y_n$，则 $y_{n-1}=f(n-1)$，本质上函数 $f(n)$ 与 $f(n-1)$ 是相同的。所以，这个函数的递归定义可表示为

$$\begin{cases} f(0)=1.41 \\ f(n)=10f(n-1)-1 \end{cases} (n=1,2,3,\cdots)$$

其中，$f(n)=10f(n-1)-1$ 为递归关系，$f(0)=1.41$ 为递归的初始条件。递归定义中这两个条件缺一不可。如果缺少最初的项，即使已知递归关系，也不能求出递归关系中包含的各项的值。类似的，由初值 $x_1=1$ 和递推关系 $x_{n+1}=\frac{1}{2}x_n+\frac{1}{x_n}$（$n$=1,2,3,4,…）确定的函数的递归定义是

$$\begin{cases} f(1)=1 \\ f(n+1)=\frac{1}{2}f(n)-\frac{1}{f(n)} \end{cases} （n=1, 2, 3, 4, \cdots）$$

一般的，递归函数包含**初始条件**和**递归表达式**，可以用如下形式表达。

$$\begin{cases} a_1=A \\ a_{n+1}=f(n,a_n) \end{cases}$$

与递归函数类似的说法，还有：

递归调用：在函数内部发出调用自身的操作。

递归算法：直接或者间接地调用自身的算法。

递归方法：通过函数或过程调用自身将问题转换为本质相同但规模较小的子问题的方法。

2. 递归算法的基本思想与构成

递归方法实际上体现了“以此类推”“用同样的步骤重复”这样的思想，是算法和程序设计中的一种重要技术。如在“$s(n)=s(n-1)+n=1+2+\cdots+n$”这个语句中，我们求 $s(n)$ 值的时候，必须先调用 $s(n-1)$；而要得到 $s(n-1)$，又必须调用 $s(n-2)$；同样，要求 $s(n-2)$ 又要调用 $s(n-3)$。依此类推，一直要递推到 $s(2)=s(1)+2$，已知 $s(1)=1$，然后代入求得 $s(2)$，从而得到 $s(3)$，这样一直可以得到 $s(n)$。

从这个递归调用过程可以看到，递归算法需要具备的两个重要条件。

（1）自身调用的语句，如 $s(n)=s(n-1)+n$；

（2）递归终止条件（即已解决的基础问题），如 $s(1)=1$。

3. 递归的逻辑过程

计算机执行递归算法程序时，总是在进行“调用”与“返回”，程序运行过程是先“调用”