



高级中学

算法与程序设计

(选学模块) (试用本) 教学参考资料



华东师范大学出版社



上海科技教育出版社



中华地图学社

◎ 高级中学

算法与程序设计

(选学模块) (试用本)

教学参考资料

华东师范大学出版社
上海科技教育出版社
中华地图学社

说 明

本教学参考资料根据上海中小学(幼儿园)课程改革委员会制定的课程方案和《上海市中小学信息科技课程标准(试行稿)》编写,供高中阶段试用。

本教材由华东师范大学主持编写,经上海市中小学教材审查委员会审查准予试用。

主编:王荣良

特约撰稿人(按姓氏笔画为序):陆晟 岳军

欢迎广大师生来电来函指出教材的差错和不足,提出宝贵意见。上海市中小学(幼儿园)课程改革委员会办公室地址:上海市陕西北路500号(邮政编码200041),联系电话:62560016(总机转)、52136338;出版社电话:62572474。

声明 按照《中华人民共和国著作权法》第二十三条中,关于“为实施九年制义务教育和国家教育规划而编写出版教科书,除作者事先声明不许使用的外,可以不经著作权人许可,在教科书中汇编已经发表的作品片段或者短小的文字作品、音乐作品或单幅的美术作品、摄影作品,但应当按照规定支付报酬,指明作者姓名、作品名称”的有关规定,我们已尽量寻找原作者支付报酬。原作者如有关于支付报酬事宜可及时与出版社联系。

前言

2004年,上海市教育委员会颁布了《上海市中小学信息科技课程标准(试行稿)》(以下简称《标准》)。《标准》指出:上海市普通中小学信息科技课程是一门以提高学生信息素养和在信息化环境下的学习能力为目标,以计算机和网络为基本载体,以学信息技术、用信息技术、懂信息技术为基本学习过程,融知识性、技能性和工具性于一体的重要的基础课程。华东师范大学出版社出版的这套《信息科技》教材就是依据《标准》,对原教材进行了全面的修订,力图以信息素养的形成为主线,以全面提高所有学生的信息素养为根本目标,从信息技术的基础知识和技能,利用信息技术发展思维、学会学习、自主探索、合作交流、解决问题,以及信息社会基本道德规范、社会责任等方面实现课程目标。修改后的教材具有以下特点:

1. 构建知识技能序,合理安排学习内容

上海市中小学信息科技课程属于12年一贯的课程群范畴,为了保证学生学习的一致性、连贯性和完整性,根据《标准》的规定、信息科技学科知识特征、学生认知规律以及社会发展需求构建知识技能序。教材修订时就预先设定课程连续主体部分的起始年段和结点年段,确定学习阶段间的衔接点和衔接方式,合理安排每一阶段的知识技能序,采用灵活多样的课程内容和形式实现课程目标。

2. 以项目活动为载体,倡导学生自主学习和相关能力的培养

根据《标准》要求,项目活动是学生能力培养的重要载体。因此,教材修改时着重加强了项目活动的设计,通过项目活动的实施来达到学生相关能力培养的目的。项目活动的设计考虑了以下几个方面:第一,项目活动的目标定位既考虑对已学知识技能的复习与巩固,更考虑学生自主学习和探索能力的培养;第二,项目活动内容选择符合学生兴趣,解决学生身边的问题;第三,项目活动的形式多样化,特别强调活动过程中的合作与交流;第四,项目活动设计体现全局观,要求学生全程参与项目活动,明确项目活动的各环节,在知识与技能运用过程中关注方法的习得;第五,项目活动体现创新性,项目活动主题适当开放,学生可以根据所在的学习环境创设项目活动内容,在体验项目活动过程的同时,发挥创造力和想象力。

3. 创设宽松和谐的学习氛围,全面渗透德育教育

《标准》提出:信息科技的课程目标之一是要使学生“具有信息化社会中的基本道德规范,应用信息技术过程中的个人自律能力,明确所承担的社会责任”。中小学信息科技课程既要培养学生必要的信息科技知识与技能,也要发展学生使用信息技术解决问题的能力,更要开展相关的道德规范和民族精神教育,在具体的情境中完善和发展学生的信息道德。因此,本教材在修订时,注意安排合适的信息安全教育、网络道德教育、国产软件应用等内容,从德育教育内容安排、道德序设计、德育教育渗透三个方面全面贯彻德育教育。

4. 关注信息科技学科核心和本质内容,体现算法思想

每门课程都有其自身的特点与要求,信息科技课程也不例外。什么是信息科技课程的核心

心内容,是一个值得探讨和研究的问题。从信息科技的发展历程和知识体系来看,算法设计与实现是信息科技课程所特有的学习内容,体现算法思想在学生生活、学习中的应用,促进学生利用信息技术解决实际问题能力的提高。本教材根据《标准》、信息科技课程的架构以及算法学习的特点,进行了如下立体化设计:在小学、初中学段以及高中统一模块中,以算法渗透为主;在高中选学模块和拓展型课程中以学习算法为主。因此,教材在小学、初中的基础型课程以及高中统一模块的学习中,主要通过概念图、项目活动的工作流程安排等环节来渗透算法思想。

5. 重视信息科技课程和其他课程的整合,多渠道培养学生信息素养

本次教材修订十分重视教材内容与其他学科知识的整合,将其他学科的内容以素材的形式融入信息科技教材中,重视信息科技在各学科领域的学习支持平台作用,集中反映信息技术在各学科领域应用的特征、方法、形式和作用,体现信息技术与各学科的整合,让学生真切体会到信息技术的时代性和工具性特征,为学生探究性学习的开展创造条件,逐步培养学生终身学习的能力。运用信息技术与学科课程整合的思想,主要体现在以下两个方面:其一,运用信息技术构建学习平台,为其他学科的学习创造条件;其二,以其他学科的内容为素材,创设学习信息技术的情景。

6. 依据教学内容和学生认知心理特征,合理设计教材栏目

教材在统一的知识能力序的架构下,依据教学内容和不同学段学生认知心理特征,合理设计不同的呈现方式。小学教材以“课”为主要结构,初中教材采用“单元”和“节”的形式,高中教材使用“章”和“节”。在栏目设计上,小学教材活泼有趣,初中教材以活动为主线,高中教材相对严谨。

本书是高中“算法与程序设计”选学模块教材的教师用书。本书以教材的“章”为序编排,每一章分为“本章教材分析”、“教学设计建议”、“评价建议”、“知识拓展”、“参考资料”五个板块,阐述高中“算法与程序设计”选学模块教材的编写思路和教学建议,为使用本教材的教师提供教学参考意见。

编者

2009年6月

目 录

第 1 章 计算机和算法

1.1 本章教材分析	1
1.2 教学设计建议	3
1.3 评价建议	8
1.4 知识拓展	9
1.5 参考资料	15

第 2 章 基本算法实例

2.1 本章教材分析	19
2.2 教学设计建议	22
2.3 评价建议	31
2.4 知识拓展	32
2.5 参考资料	37

第 3 章 程序设计基础

3.1 本章教材分析	42
3.2 教学设计建议	45
3.3 评价建议	49
3.4 知识拓展	51
3.5 参考资料	58

第 4 章 基本算法实现

4.1 本章教材分析	60
4.2 教学设计建议	62
4.3 评价建议	67
4.4 知识拓展	68
4.5 参考资料	73

第1章 计算机和算法

1.1 本章教材分析

本教材是高中信息科技选学模块。学习本模块内容,目的是进一步激发学生对计算机探索学习的兴趣,培养运用计算机解决具体问题的基础能力,养成良好的思维与程序设计基本习惯。本章作为全书的开始部分,起一个领进门的作用,内容相对较少,主要让学生理解计算机解决问题(算法设计及程序实现)的一般过程,掌握算法的概念及特点,能够用框图描述出已确定的算法。本章的标题为“计算机和算法”,就是希望教师能引导学生明白计算机与算法的关系:计算机离不开软件,软件是各种各样的程序,而程序是算法的计算机最终实现。

一、内容框架

节	小节	知 识 点	学 生 活 动
计算机解决问题的一般过程	从问题到算法	程序的组成:指令、数据	做猜价格游戏,归纳计算机解决问题的一般方法
		计算机解决问题的三个阶段:分析问题,确定解决问题的方法或软件,计算机进行处理	
	计算机与程序	程序的两种来源:应用现成软件、自己编写程序	在教师的引导下总结出计算机与程序的关系
		设计程序要考虑的两个方面:数据的存储、计算实现的过程	
算法的概念		指令的种类:输入、输出、数学运算、逻辑判断、控制转移	结合生活实例,理解算法的概念
		算法是指解决问题的具体方法和步骤	
算法的表示方法	算法的流程图表示	有穷性、确定性、可行性、有输入和输出	学习猜价格游戏算法的流程图表示
		输入与输出框、开始与结束符、处理框、连接框、流程线	
	变量及用途	如何将自然语言描述的算法通过流程图描述出来	学习求若干个数据平均值算法的流程图表示
		变量的特点	
		变量与数学公式中变量的区别	
		变量命名	
	算法的执行流程	计数器 and 累加器	完成一个算法的流程图,学习三种执行流程模式
		顺序模式的执行	
		选择模式的执行、选择的条件与执行内容	
		重复模式的执行、重复的条件与执行的内容	

二、教学目标

[知识目标]

1. 理解程序、指令、数据及它们之间的相互关系。
2. 理解算法的概念与特点。
3. 知道计算机求解问题的基本过程。
4. 知道算法的流程图表示方法。
5. 理解变量及其用处,理解算法的三种执行流程模式。

[技能目标]

1. 能用流程图描述简单的算法,能解释简单算法的执行过程和结果。
2. 能根据算法的要求选择合适的变量(变量的个数、变量命名),理解计数器、累加器。

[能力目标]

能根据实例,经历从需求分析、算法确定到编程求解等用计算机解决问题的基本过程,掌握用计算机解决问题的方法。

[情感目标]

体验计算机解决问题的特点和优势以及描述出算法的愉悦。

三、教材分析

教材从“猜价格游戏”引入“从问题到算法”,介绍了计算机程序解决问题的一般过程及算法的概念,再通过几个简单的问题介绍算法的表示方法,要求学生学会用自然语言和流程图来描述算法,引导学生探究计算机解决问题的过程。尽管高中学生已经能够利用计算机解决一些简单问题(如文章编辑、绘画和简单的多媒体处理等),但对利用计算机解决问题的完整过程还不够了解。多数人对解决问题的程序界面有所了解,但对解决问题的算法不甚了解。多数学生对程序设计语言接触较少,对基本概念和语句编写、程序调试不熟练,独立调试程序能力较差。本章通过一些实例,介绍算法概念以及计算机解决问题的基本过程,使学生对课程中所学算法知识有具体的概念并作进一步的总结和提升,为他们学习后续章节提供必要的准备。

本章作为教材的开始部分,第一节从一个课堂活动“猜商品价格”游戏入手,让学生体验人类是如何解决问题的,知道同一个问题可能有不同的解决方法,为后面引入算法的概念(算法是解决问题的具体步骤)埋下伏笔。学生了解计算机解决问题的过程和步骤,并能够总结出人类解决问题与计算机解决问题的相同点与不同点。第一节围绕课堂活动的展开,列出了一种猜商品价格方法的“指令”序列,作为实践体验,要求学生能写出第二种猜商品价格方法的“指令”序列;同时也引入了指令、程序等概念,具体而形象地说明了计算机与

程序的关系。

第二节简明扼要地介绍了算法的概念及其基本特点,目的是让学生明白“算法是解决问题的具体方法”而不是“计算的方法”,算法的特点决定了程序解决问题的特点,同时也决定了一个完整程序的几个部分。

第三节内容比较丰富,首先是介绍了算法的几种表示方法(即描述方法):自然语言描述、流程图描述、伪代码或直接用计算机程序描述,重点要求学生学会使用流程图描述算法。流程图的两种模式中只要求学生会用框图,本教材后续章节中采用流程图描述算法时使用的也都是框图。

其次,教材中通过具体例子来说明算法(程序)中变量的概念与作用,通过求矩形面积的算法和求平均值的算法来说明变量的用处、命名方法、特点以及程序变量与普通数学变量的不同。程序一方面可以通过变量来表示相应的量,即在程序运行时输入相应数据来赋给指定变量;另一方面,变量也是程序运行过程中用来记录、存储数据的,程序离不开变量。如果程序没有变量,那么再高级的计算机充其量也只是一个计算器,而现在许多计算器也都有记忆和编程功能,也能使用变量。

再次,教材中叙述了算法的三种不同的执行流程模式:顺序模式、选择(分支)模式、重复(循环)模式,这也是本章的重点。教材中将这三种模式都用流程图的方式描述出来,比较直观。结合流程图,要求学生能够理解这三种模式的执行过程,难点是选择模式与重复模式的条件判定及相应的执行内容。算法的三种不同执行流程模式在不同教材中的叫法可能有所不同,本教材中使用的是:顺序模式、选择模式、重复模式。

因此,本章围绕算法,安排了算法概念和特点,计算机、程序、算法之间的关系,算法的流程图表示,算法的三种不同执行流程模式等内容。其中,本章的重点是能用流程图表示算法,为以后各章的学习打好基础;本章的难点是理解算法的概念。

本章共安排了四个“实践体验”,第一个是“猜价格游戏”,作为算法学习的导入之用,安排在课堂上进行,学生通过游戏可以体会生活中的算法。其他三个实践体验都可以作为课外作业。学生可以在理解教学内容的基础上完成这些“实践体验”。

四、课时分配

节 次	参考课时数
第一节 计算机解决问题的一般过程	1 课时
第二节 算法的概念	1 课时
第三节 算法的表示方法	2 课时

1.2 教学设计建议

本章共分三节,依次解决三个问题:算法在计算机解决问题过程中的作用、什么是算法以及怎样描述算法。

一、计算机解决问题的一般过程

1. 教学要求

让学生通过课堂活动归纳出计算机解决问题的一般过程。

2. 教学建议

本节从一个问题实例“猜商品价格”开始,推出解决这个问题的两种方法,教师可利用这个实例提出一些问题供学生思考,为后面引出算法的概念与特点作充分准备,这样学生在活动的参与和讨论中不知不觉地了解了算法的一般概念与特点,也可为第二章中介绍的顺序查找和对分查找这两种查找算法埋下伏笔。

本节的重点就是通过活动让学生体会解决问题是需要算法的,不同的算法表达了解决问题的不同方法。

教材中的猜商品价格活动提出了两种方法,在讨论的时候,可能会有学生提出两种方法平均猜测次数或最多猜测次数的问题,教师这时可以顺势利导,提出这是两种不同的算法,本课程就是要学习一些基本的算法,为以后应用程序解决问题打下基础。算法的评估有专门的方法,应用计算机编程来解决问题首先是要找出有效的方法,然后再考虑算法的效率及优化问题。

教师可以事先编好相应的 VB 程序,由计算机来代表“B 同学”,操作计算机的为“A 同学”。第一种方法简单而无趣,最少猜测的次数为 1,最多为 1000,平均次数约 500,但它适合任意情况,如果“A 同学”只告诉“B 同学”猜测得“对”或“不对”,似乎只能用这种方法;第二种方法是不断缩小数据范围,最终猜出价格,最多猜测次数为 10(即 $\log_2 1000 + 1 = 10$),但这种方法是“A 同学”必须回答“大了”、“小了”或“对了”三者之一。两种方法各有其特点,都能够准确地“猜”出商品的价格,因而都是有效的“算法”。

本节部分概念相对比较抽象,如指令、程序,这是本节的难点。虽然这些词汇经常见到,但具体概念要讲清楚,让学生真正理解。对于指令的种类可以先作简单的介绍,等后面学习算法的概念及特点时再将两者结合起来,让学生更清楚地理解这一点:程序是实现算法的有效手段,为程序的书写而设计的指令集合完全是为了符合算法的特点和方便算法的实现。

应用计算机解决问题的三个阶段的介绍应结合人工解决问题的普遍过程:观察、分析问题,收集必要的信息,然后根据已有的知识、经验进行判断、推理,尝试按照一定的方法和步骤去解决问题。教师结合教材中的流程图简明扼要地作一个说明就可以了。

对于计算机与程序的关系,学生看了教材就能明白,教师可以从两个方面入手说明两者之间的关系:一是计算机系统包括两个部分——“硬件系统”和“软件系统”,所谓的软件系统就是各种各样的程序,现在几乎所有的计算机硬件中都固化了一些软件(程序);二是软件(程序)只能依靠硬件(具体的设备)来运行。

但这里存在的难点是:学生由于没有编写过程序,尽管教材对这些概念作了必要的叙述,但他们还是对程序、指令等概念以及计算机的存储单元缺乏直观的认识。所以,在分析存放在计算机内存中的“A 同学工作所需要的程序”时,可以忽略有关程序和指令的细节,包括变量概念,重点说明清楚几件事:其一,从“显示‘输入价格’”开始的一系列操作是有序地存放在计算机中的,以便依次读取执行;其二,计算机中既存放需要执行的指令,也预先存储需要运算的数据。

由于本节内容比较少,考虑到课时的安排,可以将本节课与下节课的内容结合起来进行教

学。教学过程可参考图 1.1。

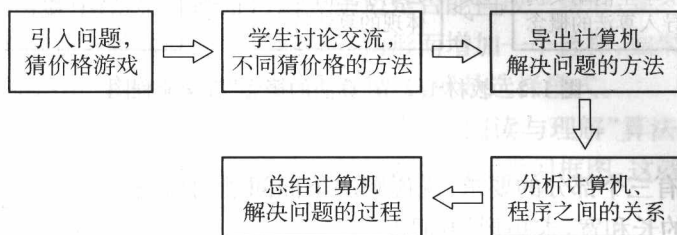


图 1.1 教材 1.1 节“计算机解决问题的一般过程”教学流程图

3. 补充练习

- (1) 列举出生活中程序化解决问题的两三个实例。
- (2) 举出同一个问题的不同解决方法例子。

二、算法的概念

1. 教学要求

由生活实例导入算法的概念并归纳出算法的特点。

2. 教学建议

本节由上一节“计算机解决问题的一般过程”的相关课堂活动展开，导出算法的概念及特点，结合具体事例，指出解决问题的方法（具体有限的步骤）就是算法，解决问题的特点也就决定了算法的特点。算法的概念与特点既要讲清楚，又要结合一些实例让学生真正理解。

本节的教学教师可采用“导学式”，结合上节的例子，提出一些问题，引导学生认识算法，理解算法的特点，让学生有一种“原来算法就在我们的身边”的感觉。

课堂引导示例：

- (1) 上一小节的猜商品价格的两种方法是不是有效的方法？解决问题的方法就是算法。
- (2) 许多算法都涉及计算，“算法就是计算的方法”这种理解对不对呢？
- (3) 两种猜商品价格方法的步骤都是有限的，如第一种方法最多 1000 次，而第二种方法最多 10 次，因此有效的算法要具备“有穷性、可行性”这样的特点。
- (4) 每一次猜商品价格时，不能给出一个模棱两可的结果，如 100 或 200, 300~600 这样不确定的结果都不符合上述两种方法。算法的一个具体步骤要可操作，具备确定性。
- (5) 针对程序设计而言的算法，最终是要交给计算机去实现的，因此要计算机处理的数据必须“告诉”计算机，这个过程称为“输入”。而计算机又是百分之百听话的，它根据程序的要求处理好数据后，没有通知它将结果告诉你的话，它是不会告诉你结果的，所以完整的算法要有“输出”，就是将处理好的结果“输送”出来。所以一个完整的算法有三部分：输入、处理、输出。

算法的概念比较抽象，教学过程中，可以将重点放在对算法特点的解释上，尽可能用实际例子来解释算法及其特点。具体实例可以选用猜价格游戏中的方法，也可以选择其他更贴近学生生活的例子，例如分析学生到图书馆查阅资料的操作过程。

所以，本节内容的教学就可以直接从上一节的学生讨论活动中导入。教学过程如图 1.2 所示。

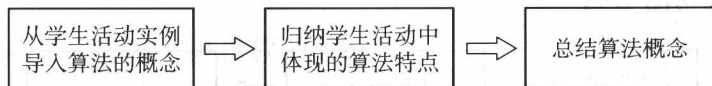


图 1.2 教材 1.2 节“算法的概念”教学流程图

3. 补充练习

根据完整算法有三个部分的要求,说说解决下面问题的算法:

- (1) 输入矩形的长和宽,求矩形的面积。
- (2) 输入三角形的三条边长,求三角形的面积。

三、算法的表示方法

1. 教学要求

学会使用自然语言和框图来描述已明确的算法。

2. 教学建议

结合具体实例,分别给出同一算法的几种描述方法。一开始可以将第二节的补充练习先用自然语言描述出来,然后再用框图描述出来,在学生理解的基础上要求将第一节中猜商品价格的两种算法描述出来。

算法的描述是本章的重点,也是以后学习算法与程序设计的基础。只有加强基础训练,熟悉算法的描述方法,才可以进行后续内容的学习。

首先可列举实例,如输入矩形的长和宽,求矩形的面积,列出操作步骤,在大家熟知方法的基础上结合算法的特点导出算法的描述,并引出记录中间数据的变量,如记录矩形的长、宽、面积可以使用三个变量如 a 、 b 、 S 等。

介绍变量和变量的用途时要围绕相应例子,让学生体会变量的重要性,至于变量的命名规则,不同的算法语言有不同的规定,可以简单地作说明,不要讲更多的“注意点”与“容易出错”的地方,强调“一般以字母开头,可长可短,中间没有空格”就够了。变量的特点是“记忆”功能,一旦给一个变量赋值了,只要不改变就能在程序运行阶段一直记住,“取之不尽,用之不竭”。教材中举了变量的两种典型基本运用:作为计数器的变量和作为累加器的变量,因为本章里并没有真正涉及程序设计语言的具体内容,赋值语句还没有作专门介绍,教师介绍赋值时一律用左向箭头“ $\leftarrow$ ”来表示,如:

$$\text{count} \leftarrow \text{count} + 1$$

$$\text{sum} \leftarrow \text{sum} + a$$

讲明它表示的含义是:先将右边式子的值计算出来,再将这个值赋给左边的变量。要求学生能够在流程图描述算法时运用这种形式。

提醒学生在今后的编程中要养成良好的习惯,就是对变量赋初值。初始化时一般要将计数器或累加器的初值置为 0。

在分析实例的操作流程时,也要不断引入各种流程图的表示符号,特别要强调处理框、输入输出框、判断框的入口和出口的不同。最后还需要对流程图符号进行归纳总结。

本节算法的三种不同执行流程模式是本节的重点,也是难点。教师可以结合生活中的事

例,用形象的比喻来加深学生对执行流程的印象,如上一节补充练习中“输入三角形的三条边长,计算并输出三角形的面积”,如果不考虑三条边是否能构成三角形,算法执行的流程模式就是顺序模式;如果考虑到三条边长是否能构成三角形而增加一个选择:如果能构成三角形则计算并输出面积,否则输出“不能构成三角形”的信息,这就是选择模式。

此外,在本节的教学中,要注意的是,需要培养学生“阅读与理解”算法的能力(自然语言、流程图、伪代码或程序描述的算法),而不是为了能画框图而学习框图,这就需要通过流程图实例来训练、体会。教学过程如图 1.3 所示。

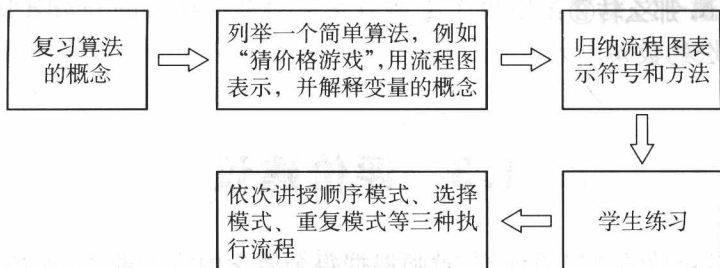


图 1.3 教材 1.3 节“算法的表示方法”教学流程图

3. 补充练习

下面的问题要求根据给出的算法自然语言相关描述画出流程图,教师可以根据学生的实际情况进行选择。

(1) 如何确定一个正整数 a 是否是质数。

自然语言描述的算法如下:

- ① 输入一个正整数 a (告诉计算机要判定哪个数);
- ② $b \leftarrow 2$;
- ③ 如果 a 能被 b 整除,那么转⑥;
- ④ b 的值增加 1;
- ⑤ 转向③;
- ⑥ 如果 $b = a$ (说明 $2 \sim a-1$ 都不能整除 a),则输出“ a 是质数”,结束;
如果 $b < a$,则输出“ a 不是质数”,结束。

(2) 求 n 个数中的最大数。

采用“擂台比武”法,基本思路是先设置一个“擂台”,在程序中用一个变量来存储最大的数,先假设最大数就是第一个数,然后从第二个数到最后一个数重复做:如果这个数比当前的最大数还要大,就取而代之,留在“擂台”。最后,“擂台”上的数就是最大数。

自然语言描述:

- ① 输入数字的个数 n ;
- ② 输入第一个数 m ;
- ③ 最大数 $max \leftarrow m$, $i \leftarrow 1$;
- ④ 输入一个数字 m ;
如果 $m > max$,那么 $max \leftarrow m$;
 $i \leftarrow i + 1$;
- ⑤ 如果 $i < n$,那么重复步骤④;

⑥ 输出最大数 max 。

(3) 欧几里得算法——求两个自然数 m 、 n 的最大公约数。

自然语言描述：

① 输入两个正整数 m 、 n ；

② $r \leftarrow m$ 除以 n 的余数；

③ $m \leftarrow n$ ；

④ $n \leftarrow r$ ；

⑤ 如果 $r > 0$ ，那么转②；

⑥ 输出最大公约数 m 。

1.3 评价建议

这里主要是针对教学效果的评价，教师根据每个教学内容的要求（如初步掌握、掌握、熟悉、熟练掌握、初步了解、了解、一般理解、理解），采用恰当的方法，检查学生的学习掌握情况，看是否达到了此要求。

下表是针对本章的知识与技能目标提出的评价建议。

节	小 节	知识评价建议	技能评价建议
计算机解决问题的一般过程	从问题到算法	了解程序的两部分组成	能够描述出计算机解决问题的一般过程
		理解计算机解决问题的三个阶段	
	计算机与程序	熟悉程序的两种来源	理解计算机与程序的关系
		初步了解设计程序要考虑的两个方面	
算法的概念		一般了解指令的种类	
		熟悉算法的概念	
算法的表示方法	算法的流程图表示	理解算法的特点	能够根据算法的特点判别算法的基本有效性
		了解流程图的两种模式	
		熟悉框图的基本图形	
	变量及用途	熟悉流程图描述算法	能够读懂流程图；能够将算法的自然语言描述转为流程图描述
		理解变量的特点	
		理解变量命名的一般原则	
		熟悉作为计数器的变量	
	算法的执行流程	熟悉作为累加器的变量	能够在算法描述中使用变量
		理解顺序模式	
		理解选择模式	
		理解重复模式	
		能够在算法中根据需要使用计数器或累加器	
		正确区分三种模式，并能对三种模式分别举例说明	

1.4 知识拓展

一、算法的定义

“算法”的中文名称出自《周髀算经》，而英文名称“algorithm”来自于9世纪波斯数学家花拉子密的名字 Al-Khwarizmi，因为花拉子密在数学上提出了算法这个概念。“算法”原为 algorism，意思是阿拉伯数字的运算法则，在18世纪演变为 algorithm。欧几里得算法被人们认为是史上第一个算法。第一个程序是 Ada Byron 于1842年为巴贝奇分析机编写求解伯努利方程的程序，因此 Ada Byron 被大多数人认为是世界上第一位程序员。因为查尔斯·巴贝奇(Charles Babbage)未能完成他的巴贝奇分析机，所以这个算法未能在巴贝奇分析机上执行。

算法的概念在计算机科学与技术领域几乎无处不在，在各种计算机系统的实现中，算法的设计往往处于核心的地位。不过计算机算法的研究受到人们如此重视是在20世纪70年代以后。具体地说，D. E. Knuth 以及 A. V. Aho、J. E. Hopcroft、J. D. Ulman 等人的著述对算法的研究起到了奠基的作用。其中，Stanford 大学著名计算机科学家 D. E. Knuth 获得了计算机领域的最高奖——图灵奖。

虽然我们每天都在与算法打交道，但要严格地指出什么是算法却不是一件容易的事。因此先把一些标准的说法介绍一下：

(1) 著名的 Webster 辞典在“algorithm”词条下指出：“算法即在有限步骤内解一个数学问题的过程，步骤中常常包括某一操作的重复。更广义地说，一个算法就是为解一个问题或实现某一目标的逐步过程。”这个定义并未与计算机相关。

(2) D. E. Knuth 给出了另一个说明：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。此外，它还应具有如下五个重要特性：

- 有穷性：一个算法必须总是在执行有穷步之后结束。
- 确定性：算法的每一步骤，必须是被确切地定义的。
- 输入：有0个或多个输入值。
- 输出：有1个或多个输出值。
- 可行性：算法中要做的运算都是相当基本的，能够精确地进行的。

有穷性说明一个无穷循环的程序不是算法；确定性则指出每一步骤要有确切含义，例如，一个菜谱就不是一个算法，因为菜谱中常有这样的步骤“加入食盐少许”，少许是不确定的；可行性的要求对于算法也是必要的，例如，算法的某一要求“如果任一偶数 $n(n > 4)$ 都可以表示为两个奇素数之和，则执行步骤 A”，“则执行步骤 A”的条件是很难确定的。

严格地说，这还不是算法的定义。一个算法的形式化定义则是基于图灵机的概念：“算法是一个对任一有效输入能够停机的图灵机。”图灵机是计算机的基本模型，有兴趣的教师可以阅读相关材料。

二、算法的评估

有时求解同一个问题常常有多种可用的算法,在一定的条件下当然要选择使用好的算法。用什么方法评估算法的好坏呢?通常使用算法复杂性这一概念来评估算法。算法复杂性又分为算法的时间复杂性和算法的空间复杂性,这两者都与问题的规模有关。求 100 个人的平均分与求 1000 个人的平均分所花的执行时间或运行空间显然有一定的差别。

算法执行时间需要通过依据该算法编制的程序在计算机上运行时所消耗的时间来度量。而度量一个程序的执行时间通常有事后统计法和事前分析估算法两种。

首先要介绍的是事后统计的方法。因为很多计算机内部都有计时功能,大部分都可以精确到毫秒级,不同算法的程序可通过一组或若干组相同的统计数据以分辨优劣。但这种方法有两个缺陷:一是必须先运行依据算法编制的程序;二是所得时间的统计量依赖于计算机的硬件、软件等环境因素,有时容易掩盖算法本身的优劣。因此人们常常采用另一种方法,即事前分析估算的方法。

一个用高级程序语言编写的程序在计算机上运行时所消耗的时间取决于下列因素:

- (1) 根据的算法选用何种策略。
- (2) 问题的规模,例如求 100 以内还是 1000 以内的素数。
- (3) 书写程序的语言,对于同一个算法,实现语言的级别越高,执行效率就越低。
- (4) 编译程序所产生的机器代码的质量。
- (5) 机器执行指令的速度。

显然,同一个算法用不同的语言实现,或者用不同的编译程序进行编译,或者在不同的计算机上运行时,效率均不相同。这表明使用绝对的时间单位去评估算法的效率是不合适的。撇开这些与计算机硬件、软件有关的因素,可以认为一个特定算法“运行工作量”的大小,只依赖于问题的规模(通常用整数量 n 表示),或者说,它是问题规模的函数。

一个算法是由控制结构(顺序、选择和重复三种)和原操作(指固有数据类型的数据操作)构成的,则算法时间取决于两者的综合效果。为了便于比较同一问题的不同算法,通常的做法是,从算法中选取一种对于所研究的问题(或算法类型)来说是基本运算的原操作,以该基本操作重复执行的次数作为算法的时间度量。

例如,在如下所示的两个 $n \times n$ 的矩阵相乘的算法中,“乘法”运算是矩阵相乘问题的基本操作。整个算法的执行时间与该基本操作(乘法)重复执行的次数 n^3 成正比,记作 $T(n) = O(n^3)$ 。下面是 BASIC 语言程序段代码:

```
for i=1 to n
  for j=1 to n do
    c(i,j)=0
    for k=1 to n
      c(i,j)=c(i,j)+a(i,k) * b(k,j)
    next k
  next j
next i
```

一般情况下,算法中基本操作重复执行的次数是问题规模 n 的某个函数 $f(n)$,算法的时间量度记作: $T(n) = O(f(n))$,它表示随着问题规模 n 的增大,算法执行时间的增长率和函数 $f(n)$ 的增长率相同,称作算法的渐进时间复杂度,简称时间复杂度。

显然,被称作问题的基本操作的原操作应是其重复执行次数和算法的执行时间成正比的原操作,多数情况下它是最深层循环内的语句中的原操作,它的执行次数和包含它的语句的频度相同。语句的频度指的是该语句重复执行的次数,例如:在下列三个程序段中:

(1) $x = x + 1$

(2) for $i = 1$ to n

$x = x + 1$

next i

(3) for $j = 1$ to n

 for $k = 1$ to n

$x = x + 1$

 next k

next i

含基本操作“ x 增 1”的语句 $x = x + 1$ 的频度分别为 1、 n 和 n^2 ,则这三个程序段的时间复杂度分别为 $O(1)$ 、 $O(n)$ 、 $O(n^2)$,分别称为常量阶、线性阶和平方阶。算法还可能呈现的时间复杂度有:对数阶 $O(\log n)$,指数阶 $O(2^n)$ 等。在 n 很大时,不同数量级时间复杂度显然有 $O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < O(2^n)$,可以看出,在设计算法时,我们应该尽可能选用多项式阶 $O(nk)$ 的算法,而不希望用指数阶的算法。

一般情况下,对一个问题(或一类算法)只需选择一种基本操作来讨论算法的时间复杂度。有时也需要同时考虑几种基本操作,甚至可以对不同的操作赋以不同权值,以反映执行不同操作所需的相对时间,这种做法便于综合比较解决同一问题的两种完全不同的算法。

由于算法的时间复杂度考虑的只是对于问题规模 n 的增长率,则在难以计算基本操作执行次数(或语句频度)的情况下,只需求出它关于 n 的增长率或阶即可。

例如,在下列程序段中:

for $i = 2$ to n

 for $j = 2$ to $i - 1$

$x = x + 1$

 next j

next i

语句 $x = x + 1$ 执行次数关于 n 的增长率为 n^2 ,它是语句频度表达式 $(n-1)(n-2)/2$ 中增长最快的一项。

一个上机执行的程序除了需要存储空间来寄存本身所用指令、常数、变量和输入数据外,也需要一些对数据进行操作的工作单元和存储一些为实现计算所需信息的辅助空间。类似于算法的时间复杂度,以空间复杂度作为算法所需存储空间的量度,记作 $S(n) = O(f(n))$,其中 n 为问题的规模(或大小)。