

高职高专

现代信息技术系列教材

数据结构

叶乃文 郑晓红 编著

Information

Technology

人民邮电出版社
www.pptph.com.cn

高职高专现代信息技术系列教材

数 据 结 构

叶乃文 郑晓红 编著

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

数据结构/叶乃文, 郑晓红编著. —北京: 人民邮电出版社, 2001.8
(高职高专现代信息技术系列教材)

ISBN 7-115-09211-7

I. 数... II. ①叶... ②郑... III. 数据结构—高等学校: 技术学校—教材
IV. TP311.12

中国版本图书馆 CIP 数据核字 (2001) 第 047715 号

高职高专现代信息技术系列教材

数 据 结 构

◆ 编 著 叶乃文 郑晓红
责任编辑 潘春燕

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京朝阳隆昌印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16
印张: 11.5

字数: 264 千字

2001 年 8 月第 1 版

印数: 1-6 000 册

2001 年 8 月北京第 1 次印刷

ISBN 7-115-09211-7/TP·2156

定价: 16.00 元

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

内 容 提 要

本书根据高职及大专层次的《数据结构》教学大纲，用通俗的语言、由浅入深的方式讲述了数据结构的基本概念、线性表、栈和队列、串和数组、树形结构、图结构、查找、排序和文件等内容，其中列举了大量的例题及各种数据结构的操作算法，在每章的后面还给出了小结及深浅适度的练习题。与其他教材相比较，本书内容系统、全面，讲解理论概念通俗易懂，选题难度适中，算法描述语言选用 C 语言，并给出大量的典型算法实现。读者通过本书的学习，能够比较轻松地掌握《数据结构》的基本内容，为后续课程的学习打下一个良好的基础。

本书既可以作为计算机相关专业的高职、大专及业余大学的教材，也可以作为计算机爱好者自学数据结构的参考书。

高职高专现代信息技术系列教材

编 委 会 名 单

主 编 高 林

副 主 编 张强华

委 员 （以姓氏笔画为序）

吕新平 林全新 郭力平 程时兴

丛 书 前 言

江泽民总书记在十五大报告中提出了培养数以亿计高素质的劳动者和数以千万计专门人才的要求,指明了高等教育的发展方向。只有培养出大量高素质的劳动者,才能把我国的人数优势转化为人力优势,提高全民族的竞争力。因此,我国近年来十分重视高等职业教育,把高等职业教育作为高等教育的重要组成部分,并以法律形式加以约束与保证。高等职业教育由此进入了蓬勃发展时期,驶入了高速发展的快车道。

高等职业教育有其自身的特点。正如教育部“面向 21 世纪教育振兴行动计划”所指出的那样,“高等职业教育必须面向地区经济建设和社会发展,适应就业市场的实际需要,培养生产、管理、服务第一线需要的实用人才,真正办出特色。”因此,不能以本科压缩和变形的形式组织高等职业教育,必须按照高等职业教育的自身规律组织教学体系。为此,我们根据高等职业教育的特点及社会对教材的普遍需求,组织高等职业学校有丰富教学经验的老师,编写了这套《高职高专现代信息技术系列教材》。

本套教材充分考虑了高等职业教育的培养目标、教学现状和发展方向,在编写中突出了实用性。本套教材重点讲述目前在信息技术行业实践中不可缺少的、广泛使用的、从业人员必须掌握的实用技术。即便是必要的理论基础,也从实用的角度、结合具体实践加以讲述。大量具体操作步骤、许多实践应用技巧、接近实际的实训材料保证了本套教材的实用性。

在本套丛书编写大纲的制定过程中,广泛收集了高等职业学院的教学计划,调研了多个省市高等职业教育的实际,反复讨论和修改,使得编写大纲能最大限度地符合我国高等职业教育的要求,切合高等职业教育实际。

在选择作者时,我们特意挑选了在高等职业教育一线的优秀骨干教师。他们熟悉高等职业教育的教学实际,并有多年的教学经验;其中许多是“双师型”教师,既是教授、副教授,同时又是高级工程师、认证高级设计师;他们既有坚实的理论知识,很强的实践能力,又有较多的写作经验及较好的文字水平。

目前我国许多行业开始实行劳动准入制度和职业资格制度,为此,本套教材也兼顾了一些证书考试(如计算机等级考试),并提供了一些具有较强针对性的训练题目。

对于本套教材我们将提供教学支持(如提供电子教案),同时注意收集本套教材的使用情况,不断修改和完善。

本套教材是高等职业学院、高等技术学院、高等专科学院教材。适用于信息技术的相关专业,如计算机应用、计算机网络、信息管理、电子商务、计算机科学技术、会计电算化等。也可供优秀职高学校选作教材。对于那些要提升自己应用技能或参加一些证书考试的读者,本套教材也不失为一套较好的参考书。

最后,恳请广大读者将本套教材的使用情况及各种意见、建议及时反馈给我们,以便在今后的工作中,不断改进和完善。

关于本书

1. 本书概览

《数据结构》是计算机相关专业的一门重要的专业基础课，主要研究计算机加工对象的逻辑结构、在计算机中的表示形式以及实现各种基本操作的算法。它是学习操作系统、编译原理、数据库原理等计算机专业核心课程的基础，掌握好本书内容，是学习计算机其他相关课程的必备条件。

这本书根据高职及大专层次的《数据结构》教学大纲，用通俗的语言、由浅入深的方式讲述了数据结构的基本概念、线性表、栈和队列、串和数组、树形结构、图结构、查找、排序和文件等内容，其中列举了大量的例题及各种数据结构的操作算法。相信读者通过本书的学习，能够比较轻松地掌握《数据结构》的基本内容，为后续课程的学习打下一个良好的基础。这也正是我们编写这本教材的初衷。

2. 学生如何使用本书

学生在学习《数据结构》时，常常感到有一定的困难。他们对一些比较抽象的概念和算法的设计思路比较难以接受，有些同学忽视对概念以及各种数据结构之间的相互关系的理解，而是死记硬背每个算法，结果越学越觉得内容繁多，最终不会得到良好的效果。我们建议学生在使用这本教材学习的时候，注意对书中介绍的基本概念进行反复地揣摩，利用给出的例子加深对其含义的体会和理解。对书中给出的所有算法要逐一读懂，并能够举一反三。注意每个算法之间的相互关系，最好能够将其其中的一些典型算法上机实现，从而加深理解程度，最终达到利用各种数据结构的基本操作完成比较复杂的操作的目的。

3. 教师如何使用本书

本书使用的对象是高职、大专层次的学生，因此，在教学内容和难易程度上与本科生教学有所区别。在讲述基本概念及具体算法时，应尽可能地实例化，用通俗的语言及由浅入深的方式诱导学生，使学生较容易地理解《数据结构》课程中比较抽象的概念，并通过举一反三的教学方法，最终使学生能够达到熟练掌握这门课程主要内容的目的。

目 录

第1章 数据结构基础概论	1
1.1 数据结构研究的主要内容	1
1.2 基本概念和术语	4
1.3 算法	4
1.3.1 算法的概念	4
1.3.2 算法的描述	5
1.3.3 算法的评价	7
本章小结	8
练习题	9
第2章 线性表	10
2.1 线性表的定义和基本操作	10
2.1.1 线性表的定义	10
2.1.2 线性表的基本操作	11
2.2 线性表的顺序存储结构	12
2.2.1 线性表的顺序存储结构及其特点	12
2.2.2 典型操作的算法实现	13
2.3 线性表的链式存储结构	16
2.3.1 线性表的链式存储结构及其特点	16
2.3.2 典型操作的算法实现	18
2.3.3 循环链表	22
2.3.4 双向循环链表	23
2.4 线性表的应用举例	25
2.4.1 使用顺序存储结构的例子	26
2.4.2 使用链式存储结构的例子	27
本章小结	29
练习题	29
第3章 栈和队列	31
3.1 栈	31
3.1.1 栈的定义	31
3.1.2 栈的顺序存储	32
3.1.3 栈的链式存储	34
3.1.4 栈的应用举例	35

3.2 队列	38
3.2.1 队列的定义	38
3.2.2 队列的顺序存储	39
3.2.3 队列的链式存储	44
3.2.4 队列的应用举例	46
本章小结	47
练习题	47
第4章 串和数组	48
4.1 串	48
4.1.1 串的定义和基本运算	48
4.1.2 串的存储结构	50
4.2 数组	53
4.2.1 数组的定义和基本运算	53
4.2.2 数组的存储结构	54
4.2.3 矩阵的压缩存储	56
本章小结	62
练习题	62
第5章 树和二叉树	64
5.1 树	64
5.1.1 树的定义和基本运算	64
5.1.2 树的存储结构	66
5.2 二叉树	70
5.2.1 二叉树的定义和基本运算	70
5.2.2 二叉树的性质	71
5.2.3 二叉树的存储结构	74
5.2.4 遍历二叉树	76
5.2.5 典型二叉树的操作算法	83
5.2.6 树、森林与二叉树的转换	87
5.3 哈夫曼树及其应用	90
5.3.1 哈夫曼树的定义	90
5.3.2 判定树	93
5.3.3 前缀编码	94
本章小结	96
练习题	96

第 6 章 图	98
6.1 图的定义	98
6.1.1 图的定义	98
6.1.2 图的基本操作	100
6.2 图的存储结构	101
6.2.1 邻接矩阵	101
6.2.2 邻接表	102
6.3 图的遍历	105
6.3.1 深度优先遍历	105
6.3.2 广度优先遍历	107
6.4 最小生成树问题	109
6.4.1 图的生成树和森林	109
6.4.2 最小生成树	110
6.5 拓扑排序问题	113
本章小结	116
练习题	116
第 7 章 查找	118
7.1 基本概念	118
7.2 静态查找	119
7.2.1 顺序查找	119
7.2.2 折半查找	121
7.3 动态查找	123
7.3.1 二叉排序树	124
7.3.2 二叉排序树的查找	124
7.3.3 二叉排序树的插入	126
7.3.4 二叉排序树的删除	128
7.4 哈希表	129
7.4.1 哈希表的概念	129
7.4.2 哈希函数的构造	129
7.4.3 解决冲突的方法	130
7.4.4 哈希表查找及其分析	130
本章小结	131
练习题	132
第 8 章 排序	133
8.1 基本概念	133
8.2 插入排序	134

8.2.1 直接插入排序	134
8.2.2 希尔排序	136
8.3 交换排序	139
8.3.1 冒泡排序	139
8.3.2 快速排序	143
8.4 选择排序	147
8.4.1 简单选择排序	147
8.4.2 堆排序	149
8.5 归并排序	154
8.5.1 归并排序的基本思想	154
8.5.2 归并排序过程示例	154
8.5.3 归并排序算法	154
8.6 基数排序	156
8.6.1 基数排序的基本思想	156
8.6.2 基数排序示例	159
8.6.3 链式基数排序算法	160
本章小结	162
练习题	163
第9章 文件	164
9.1 文件的基本概念	164
9.2 文件的操作	165
9.3 文件的组织	165
本章小结	166
练习题	166

第 1 章 数据结构基础概论

自从 1946 年世界上第一台计算机诞生以来, 计算机硬件和软件的发展速度震惊了世人, 人们不得不对计算机这个人类的得力助手给予一次又一次新的认识。众所周知, 计算机在发展初期, 主要是用来进行数值的计算, 它可以帮助人们处理可能需要花费很多人力、物力才能完成的海量数值计算, 其计算速度及准确程度令人刮目相看。但是, 那时所处理的数据之间基本上没有什么复杂的关系, 形式也比较单一。所以, 从事计算机软件的人员把大量的精力放在了提高程序设计的技巧上, 使之加快计算速度, 减少存储空间的占用率, 而并没有重视如何利用计算机组织大量的数据。随着计算机软、硬件的不断发展, 其应用领域也在不断地扩展, 所处理的数据渐渐地复杂起来, 这时的计算机开始逐步涉足于非数值性处理。所谓非数值性处理是指: 所处理的数据之间存在着某种特定的关系, 计算机在处理它们时需要对这些关系进行表示、维护, 其主要的操作也不再是单纯的计算, 而是转向对大量的数据进行组织、管理和检索等等。正是由于计算机应用领域的这种变迁, 吸引了大批的计算机人士, 开展对计算机所加工的各种数据关系以及各种数据在计算机中的表示、处理的方法进行研究。这样就逐步形成了这门计算机专业的核心专业基础课程——《数据结构》, 它是学习许多其他计算机专业课程的基础。

在本章中将主要介绍以下内容:

- 数据结构研究的主要内容
- 数据结构中涉及的基本概念
- 算法的概念、描述方法以及评价标准

1.1 数据结构研究的主要内容

在我们大踏步地跨入新世纪的时候, 计算机也同人类共度了半个多世纪。计算机迅猛发展的速度为世人有目共睹。在当今人类社会中, 计算机无处不有, 无处不用。在学校, 我们应用计算机管理学籍档案; 在图书馆, 我们应用计算机管理图书、资料信息; 在铁路部门和航空公司, 我们应用计算机管理车票、机票; 在医院, 我们应用计算机管理病历档案等等。这些应用的共同特点是所处理的数据量大且具有一定的关系, 对其操作不再是单纯的数值计算, 而更多地是需要对这些数据进行组织、管理和检索等等。下面我们列举几个这方面的例子。

【例1-1】 学籍档案管理

一个实际的学籍档案管理系统应包含学生的所有信息。比如：学生的基本情况、学生的学习成绩、学生的升留级情况、学生的获奖或受处分情况等等。为方便起见，我们在此简化为只包含学生基本情况中的部分信息：学号、姓名、性别、出生年月……并将其表示成表 1-1 的表格形式。

表 1-1 学生基本情况

学 号	姓 名	性 别	出生年月	
99070101	李 军	男	80. 12	
99070102	王颜霞	女	81. 2	
99070103	孙 涛	男	80. 9	
99070104	单晓宏	男	81. 3	
.....				

其中，每个学生的信息占据一行，所有学生的信息按学号顺序依次排列构成一张表格。可以看到，在表格中每个学生的信息依据学号的大小存在着一种前后关系，即学号较小者在前、学号较大者在后。这就是我们所说的线性结构。对它的操作通常是插入某个学生的信息、删除某个学生的信息、更新某个学生的信息、按条件检索某个学生的信息等等。

【例1-2】 输出 n 个对象的全排列

大家都知道, n 个对象共有 $n!$ 种排列方式。我们假设将 n 个对象分别用 $1, 2, 3, \dots, n$ 编号表示, 求解这个问题的方法可以描述为: 已知 1 个对象的全排列为 1 种, 即 1 ; 在 1 的前后分别插入 2 , 就得到了两个对象的全排列, 21 和 12 ; 再对两个对象的两种排列分别在 3 个插入位置上插入 3 , 就得到了 3 个对象的全排列: $321, 231, 213, 312, 132, 123$; 依此类推, 若已知 $k-1$ 个对象的全排列, 将 k 插入到每种排列中的 k 个插入位置上,

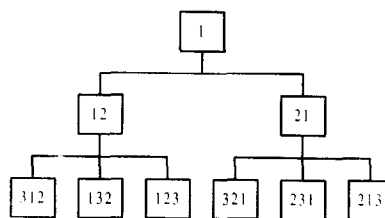


图 1-1 3 个对象的全排列过程

就得到了 k 个对象的全排列，直到 $k=n$ 为止。我们可以将这个过程表示成图 1-1 所示的形式。

由此可以看出：在求解这个问题的过程中，所处理的数据之间具有层次关系，第一层是1个对象的全排列；第二层是2个对象的全排列，它是以第一层为基础得到的；第三层是3个对象的全排列，它是以第二层为基础得到的。我们将这种层次结构称为树形结构。对它的操作有建立树形结构，输出最低层结点内容等等。

【例1-3】 制定教学计划

在制定教学计划时,需要考虑各门课程的开设顺序。有些课程需要先导课程,有些课程则不需要,而有些课程又是其他课程的先导课程。比如,某校计算机专业必修课程的开设

情况如表 1-2 所示。

表 1-2

计算机专业学生的必修课程

课程编号	课程名称	需要的先导课程编号
c_1	程序设计基础	无
c_2	离散数学	c_1
c_3	数据结构	c_1, c_2
c_4	汇编语言	c_1
c_5	算法分析与设计	c_3, c_4
c_6	计算机组成原理	c_{11}
c_7	编译原理	c_5, c_3
c_8	操作系统	c_3, c_6
c_9	高等数学	无
c_{10}	线性代数	c_9
c_{11}	普通物理	c_9
c_{12}	数值分析	c_9, c_{10}, c_1

下面我们将这些课程的先后关系用图 1-2 的形式描述。

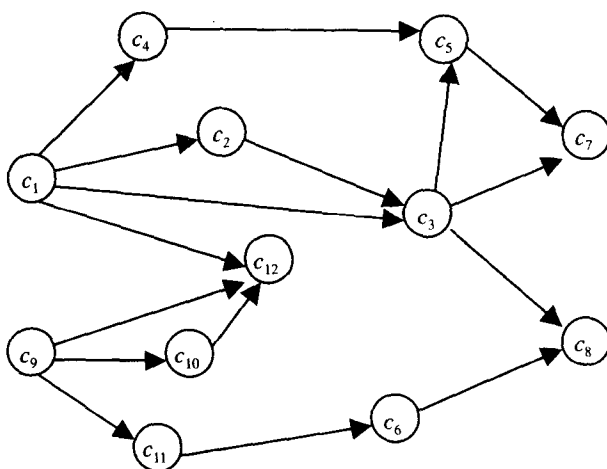


图 1-2 计算机专业必修课程开设先后关系

其中，用圆圈代表课程，圆圈中的字符代表课程的编号，我们将其称之为顶点。若某门课程在开设之前必须先修另一门课程，则在这两个顶点之间就有一条由先导课程指向该课程的有向边。这种结构被称为图形结构。通过对它的操作，就可以得到一种课程的线性排列顺序，该顺序一定能够满足先导课程在后序课程前开设的要求。

从上面 3 个例子可以看出：计算机的操作对象和操作形式不再是单纯的数值计算，而更多地是对这些具有一定关系的数据进行组织、管理的非数值性处理。要使计算机能够更有效地进行这些非数值性处理，就必须弄清楚这些操作对象的特点、在计算机中的表示方式以

及各个操作的具体实现手段。这些就是《数据结构》这门课程研究的主要内容。

1.2 基本概念和术语

在本节中，我们将对后续章节中用到的有关概念和术语给予明确的定义，以免在进一步地学习过程中，产生理解上的误差。

数据 是对客观事物的符号表示。在计算机科学中其含义是指所有能够输入到计算机中并被计算机程序处理的符号集合。例如，求解一元二次方程 $ax^2+bx+c=0$ 程序处理的对象是整型或浮点数值；学生档案管理系统中处理的对象是表示学生姓名、家庭住址信息的字符串和记录学生各门课程考试成绩的整型数值等等。随着计算机的不断发展，数据的概念也在不断地扩展，现在，音频、视频、图像等媒体都可以通过数字化编码后成为计算机加工处理的对象。

数据元素 是数据集合中的一个实体，是计算机程序中加工处理的基本单位。例如，在上述所说的学籍档案管理系统中，通常将每个学生的信息作为处理的基本单位。数据元素按其组成可分为简单型数据元素和复杂型数据元素。简单型数据元素由一个数据项组成，所谓数据项就是数据中不可再分割的最小单位，比如，学生信息中的学号、姓名、性别等等。复杂型数据元素由多个数据项组成，它通常携带着一个概念的多方面信息，比如，描述一本书的信息应该有书号、书名、作者、出版社名称等等。

数据结构 简单地说，就是相互之间存在一种或多种特定关系的数据元素的集合。我们在上一节中列举的 3 个例子都具有这个特点，即其中的数据元素不是孤立存在的，而是相互之间存在着某种关系，很多人又将这里所说的关系称为结构。常见的数据结构有线性结构、树形结构和图形结构。

逻辑结构 数据结构中所说的“关系”实际上是指数据元素之间的逻辑关系，又称为逻辑结构。比如，我们在前面提到的学生信息按学号顺序排列的前后关系，在求解 n 个对象的全排列时数据之间形成的上下层关系，以及开设各门课程的先后关系等都指的是逻辑关系，即逻辑结构。

存储结构（物理结构） 是指数据结构在计算机存储器中的具体实现。与孤立的数据元素表示形式不同，数据结构中的数据元素不但要表示其本身的实际内容，还要表示清楚数据元素之间的逻辑结构。常见的存储结构有顺序存储结构和链式存储结构。顺序存储结构的特点是借助于数据元素的相对存储位置来表示数据元素之间的逻辑结构，而链式存储结构的特点是借助于指示数据元素地址的指针表示数据元素之间的逻辑结构。

1.3 算 法

1.3.1 算法的概念

算法是解决某个特定问题的一种方法或一个过程。计算机对数据的操作可以分为数值

性和非数值性两种类型。在数值性操作中主要进行的是算术运算；而在非数值性操作中主要进行的是检索、排序、插入、删除等等。算法也分为两大类：数值性操作算法和非数值性操作算法。比如，求解代数方程、微分积分方程的算法都属于数值性算法。而像组织、管理数据的数据检索算法、排序算法、遍历算法等等就属于非数值性算法。数据结构中所讨论的算法都属于非数值性算法，它是针对一种逻辑结构，在某种存储结构下，进行某个特定操作的具体实现过程的描述。

我们在这里所说的算法准确地说应该是“计算机算法”。因为当我们需要利用计算机解决一个实际问题时，首先应该通过对问题的详细分析，抽象出相应的数学模型；然后确定所使用的数据结构，并在此基础上设计对此数据结构实施各种操作的算法；最终，需要选用某种语言将算法转换成程序，并在计算机上运行这些程序以得到我们所希望的结果。因此，设计的算法应该具有下列5个特性：

- (1) 有穷性：一个算法必须在执行有穷步之后结束。
- (2) 确定性：算法中的每一步，必须有确切的含义，使他人理解时不会产生二义性。
- (3) 可行性：算法中描述的每一步操作都可以通过已有的基本操作执行有限次来实现。
- (4) 输入：一个算法应该有零个或多个输入。这里所说的输入包括：由输入设备人工输入的数据、通过参数传递的数据、读取文件获取的数据、利用随机数发生器产生的数据等等。
- (5) 输出：一个算法应该有一个或多个输出。这里所说的输出是指与输入有某种特定关系的量。输出形式包括通过参数或函数返回值传输结果数据，在输出设备上直接显示结果数据，将结果数据写入外部文件等等。

下面我们列举一个算法的例子。

按从小到大的顺序重新排列 x 、 y 、 z 三个数值的内容。

解决这个问题的算法可以描述为：

- (1) 输入 x 、 y 、 z 三个数值；
- (2) 从三个数值中挑选出最小者，并换到 x 中；
- (3) 从 y 、 z 中挑选出较小者，并换到 y 中；
- (4) 输出排序后的结果。

可以看出，这个算法具备上述5个特性。

1.3.2 算法的描述

选择描述数据结构和算法的语言工具是十分重要的问题。选择的准则主要有以下3条：

- (1) 该语言应该具有描述数据结构和算法的基本功能；
- (2) 该语言应该尽可能地简捷，以便于掌握、理解；
- (3) 使用该语言描述的算法应该能够比较容易地转换成任何一种程序设计语言。

目前，大多数人比较认同以一种程序设计语言为基础，经过一定的取舍形成一种比较简练、方便的算法描述语言，其目的主要是为了更好地体现算法的实现过程，而免去了由于语言本身繁多的语法规则带来的困扰。本书也将如此，并以广泛使用的 C 语言为基础，构造一种被称为“类 C”的描述语言来实现对各种数据结构及算法的操作描述。

本书采用的“类 C”描述语言是通过对 C 语言进行精心筛选保留的一个核心子集，并为了便于描述，又做了若干扩展修改，从而，增强了语言的描述功能。下面对其做一个简要

的说明。

1. 预定义常量及类型

```
#define TRUE 1
#define FALSE 0
#define OK 1
#define ERROR 0
#define OVERFLOW -1
```

数据元素被约定为 **EntryType** 类型，用户需要根据具体情况，自行定义该数据类型。

2. 算法描述为以下的函数形式

函数类型 函数名 (函数参数表)

```
{
    语句序列;
}
```

为了简化函数的书写，提高算法描述的清晰度，我们规定：除函数参数表中的参数需要说明数据类型外，函数中使用的局部变量可以不做变量说明，必要时给出相应的注释即可。另外，在书写算法时，应该养成对重点语句段落添加注解的良好习惯。

3. 在算法描述中可以使用的赋值语句形式

简单赋值 变量名 = 表达式;

串联赋值 变量名 1 = 变量名 2 = ... = 变量名 n = 表达式;

成组赋值 (变量名 1, ..., 变量名 n) = (表达式 1, ..., 表达式 n);

结构赋值 结构名 1 = 结构名 2;
结构名 = (值 1, 值 2, ..., 值 n);

条件赋值 变量名 = 条件表达式 ? 表达式 1: 表达式 2;

交换赋值 变量名 1 $\leftrightarrow$ 变量名 2;

4. 在算法描述中可以使用的选择结构语句形式

条件语句 1 if (表达式) 语句;

条件语句 2 if (表达式) 语句;
else 语句;

开关语句 1 switch (表达式) {
case 值 1: 语句序列 1; break;
case 值 2: 语句序列 2; break;
.....
case 值 n: 语句序列 n; break;
default: 语句序列 n+1;
}

开关语句 2 switch {
case 条件 1: 语句序列 1; break;
case 条件 2: 语句序列 2; break;