

21世纪高职高专规划教材

计算机专业基础系列



# 数据结构

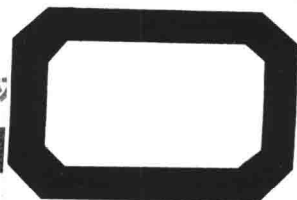
邓文华 主编

清华大学出版社



21 世纪高职高专规划教材

计算机专业基础系列



# 数据结构

邓文华 主编



清华大学出版社  
北京

## 内 容 简 介

本书对常用的数据结构做了系统的介绍,力求概念清晰,注重实际应用。全书共分8章,依次介绍了数据结构的基本概念、线性表、栈和队列、串和数组、树结构和图结构,以及查找和排序等基本运算。全书以C语言作为算法描述语言,且每章后均列举了典型应用实例,并配有电子课件,供教师教学使用。

本书配套出版了《数据结构实验与实训教程》,便于教学实验、课程设计及复习应考。本书主要面向高职高专院校计算机专业的学生,也可以作为大学非计算机专业的选修课教材和计算机应用技术人员自学参考书。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

数据结构/邓文华主编. —北京:清华大学出版社,2004.8

(21世纪高职高专规划教材. 计算机专业基础系列)

ISBN 7-302-08978-7

I. 数… II. 邓… 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(2004)第066358号

出版者:清华大学出版社

<http://www.tup.com.cn>

社总机:010-62770175

地址:北京清华大学学研大厦

邮编:100084

客户服务:010-62776969

组稿编辑:束传政

文稿编辑:田梅

印装者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开本:185×230 印张:12 字数:242千字

版次:2004年8月第1版 2005年1月第2次印刷

书号:ISBN 7-302-08978-7/TP·6350

印数:5001~10000

定价:18.00元

---

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103或(010)62795704

# 前 言

## 数据结构

数据结构是计算机程序设计的重要理论基础,该课程是计算机及其应用专业的一门重要基础课程和核心课程。它不仅是计算机软件专业课程的先导,而且也逐渐为其他工科类专业所重视。

本教材主要面向高职高专院校计算机专业的学生。教材内容结构力求体现以应用为主体,强调理论知识的理解和运用,实现专科教学以实践体系及技术应用能力培养为主的目标。

本书共分8章。第1章叙述数据、数据结构和算法等基本概念。第2章~第6章分别讨论了线性表、栈和队列、串和数组、树和图等基本数据结构及其应用。第7章和第8章分别讨论了查找和排序的各种实现方法及其应用。

本教材具有以下特点。

(1) 基础理论知识的阐述由浅入深、通俗易懂。内容的组织和编排以应用为主线,省略了一些理论推导和数学证明过程,淡化了算法的设计分析和复杂的时空分析。

(2) 各章(除第1、8章外)都配有应用举例,列举分析了很多实用的例子,且大多数算法都直接给出了相应的C语言程序,以便上机练习和实践。

(3) 为了便于复习和掌握每章的重点,每章的起始处都给出了要点,并在每章结尾处给出了小结。

(4) 本教材配有相应的PowerPoint演示文稿。

本教材推荐课时数约为64学时,上机时数约为20学时(有需要的院校请与我们联系:gzzz@tup.tsinghua.edu.cn)。本书由邓文华副教授主编,编写了该书的第1~4章,第5、6章由李益明老师编写,第7、8章由戴大蒙老师编写。全书由邓文华老师最后修改定稿。浙江大学计算机学院孔繁胜教授审稿。

编写高职高专计算机专业教材是一项新的尝试,因编写时间紧迫,难免存在疏漏,敬请读者批评指正。

编 者

2004年7月

# 目 录

## 数据结构

<b>第 1 章 绪论</b> .....	1
本章要点.....	1
1.1 数据结构概念.....	1
1.1.1 为什么要学习数据结构.....	1
1.1.2 有关概念和术语.....	4
1.2 算法描述.....	7
1.2.1 算法特性.....	7
1.2.2 算法描述.....	8
1.3 算法分析.....	9
小结.....	10
习题.....	11
<b>第 2 章 线性表</b> .....	12
本章要点.....	12
2.1 线性表的逻辑结构.....	12
2.1.1 线性表的定义.....	12
2.1.2 线性表的基本操作.....	13
2.2 线性表的顺序存储及操作实现.....	14
2.2.1 顺序表.....	14
2.2.2 顺序表的基本操作实现.....	16
2.2.3 顺序表应用举例.....	20
2.3 线性表的链式存储和运算实现.....	22
2.3.1 单链表.....	22
2.3.2 单链表上基本运算的实现.....	24

2.3.3	循环链表 .....	30
2.3.4	双向链表 .....	31
2.4	应用举例 .....	33
	小结 .....	35
	习题 .....	36
<b>第3章</b>	<b>栈和队列 .....</b>	<b>38</b>
	本章要点 .....	38
3.1	栈 .....	38
3.1.1	栈的定义和基本运算 .....	38
3.1.2	栈的存储实现和运算实现 .....	39
3.2	栈的应用举例 .....	42
3.3	队列 .....	47
3.3.1	队列的定义和基本运算 .....	47
3.3.2	队列的存储实现和运算实现 .....	48
3.4	队列应用举例 .....	54
	小结 .....	55
	习题 .....	55
<b>第4章</b>	<b>串和数组 .....</b>	<b>58</b>
	本章要点 .....	58
4.1	串 .....	58
4.1.1	串的基本概念 .....	58
4.1.2	串的基本运算 .....	59
4.1.3	串的存储和基本运算实现 .....	60
4.2	数组 .....	63
4.2.1	数组的逻辑结构 .....	63
4.2.2	数组的内存映像 .....	64
4.2.3	稀疏矩阵 .....	65
4.3	应用举例 .....	69
	小结 .....	70
	习题 .....	70

<b>第 5 章 树和二叉树</b> .....	72
本章要点 .....	72
5.1 树的概念和基本操作 .....	72
5.1.1 树的定义和相关术语 .....	72
5.1.2 树的基本操作 .....	74
5.2 二叉树 .....	75
5.2.1 二叉树的基本概念 .....	75
5.2.2 二叉树的主要性质 .....	76
5.2.3 二叉树的基本操作与存储实现 .....	77
5.2.4 二叉树的遍历 .....	81
5.3 树和森林 .....	84
5.3.1 树的存储 .....	84
5.3.2 树、森林与二叉树的转换 .....	87
5.3.3 树和森林的遍历 .....	89
5.4 最优二叉树——哈夫曼树 .....	91
5.4.1 哈夫曼树的基本概念 .....	91
5.4.2 哈夫曼树的构造算法 .....	92
5.4.3 哈夫曼树在编码问题中的应用 .....	94
5.5 应用举例 .....	96
小结 .....	99
习题 .....	99
<b>第 6 章 图</b> .....	102
本章要点 .....	102
6.1 图的基本概念 .....	102
6.1.1 图的定义和术语 .....	102
6.1.2 图的基本操作 .....	106
6.2 图的存储表示 .....	106
6.2.1 邻接矩阵 .....	107
6.2.2 邻接表 .....	109
6.3 图的遍历 .....	111
6.3.1 深度优先搜索 .....	111
6.3.2 广度优先搜索 .....	113

6.4	图的应用 .....	114
6.4.1	最小生成树 .....	114
6.4.2	最短路径 .....	118
6.4.3	拓扑排序 .....	121
6.5	应用举例 .....	125
	小结 .....	129
	习题 .....	130
<b>第7章</b>	<b>查找 .....</b>	<b>133</b>
	本章要点 .....	133
7.1	基本概念和术语 .....	133
7.2	静态查找表 .....	135
7.2.1	静态查找表结构 .....	135
7.2.2	顺序查找 .....	135
7.2.3	有序表的折半查找 .....	136
7.2.4	分块查找 .....	139
7.3	动态查找表 .....	140
7.4	哈希表 .....	144
7.4.1	哈希表和哈希方法 .....	144
7.4.2	常用的哈希函数 .....	145
7.4.3	冲突处理方法 .....	147
7.4.4	哈希表的查找及其分析 .....	149
7.5	应用举例 .....	150
	小结 .....	158
	习题 .....	158
<b>第8章</b>	<b>排序 .....</b>	<b>160</b>
	本章要点 .....	160
8.1	基本概念 .....	160
8.2	简单排序方法 .....	161
8.2.1	直接插入排序 .....	161
8.2.2	冒泡排序 .....	163
8.2.3	简单选择排序 .....	164
8.3	快速排序 .....	165



8.4 堆排序 .....	168
8.5 归并排序 .....	171
8.6 基数排序 .....	173
8.6.1 多关键码排序 .....	173
8.6.2 链式基数排序 .....	174
8.7 排序方法的比较和讨论 .....	176
小结 .....	177
习题 .....	178
<b>参考文献</b> .....	<b>179</b>

## 绪 论

计算机科学是一门研究数据表示和数据处理的科学。数据是计算机化的信息,是计算机可以直接处理的最基本和最重要的对象。无论是进行科学计算、数据处理、过程控制,还是对文件的存储和检索及数据库技术的应用,都是对数据进行加工处理的过程。因此,要设计出一个结构好、效率高的程序,必须研究数据的特性及数据间的相互关系及其对应的存储表示,并利用这些特性和关系设计出相应的算法和程序。

### 本章要点

- (1) 数据、数据元素、数据结构、数据的逻辑结构与物理结构的概念以及逻辑结构与物理结构间的关系。
- (2) 数据类型的概念。
- (3) 算法的定义、特性以及算法的时间复杂度和空间复杂度。
- (4) 用 C 语言描述算法的方法,使用 C 语言编写算法程序。

## 1.1 数据结构概念

数据结构是计算机科学与技术专业的专业基础课,是十分重要的核心课程。所有的计算机系统软件和应用软件都要用到数据结构。因此,要想更好地运用计算机来解决实际问题,仅掌握几种计算机程序设计语言是难以应对众多复杂的课题的。要有效地使用计算机,充分发挥计算机的性能,还必须学习和掌握好数据结构的有关知识。具备数据结构的扎实基础,对于学习计算机专业的其他课程,如操作系统、编译原理、数据库管理系统、软件工程、人工智能等都是十分有益的。

### 1.1.1 为什么要学习数据结构

在计算机发展的初期,人们使用计算机的目的主要是处理数值计算问题。使用计算

机解决一个具体问题时,一般需要经过下列步骤。首先要从具体问题中抽象出一个适当的数学模型,然后设计或选择一个求解此数学模型的算法,最后编出程序进行调试、测试,直至得到最终的解答,如图 1-1 所示。例如,求解梁架结构中应力的数学模型的线性方程组,该方程组可以使用迭代算法来求解。

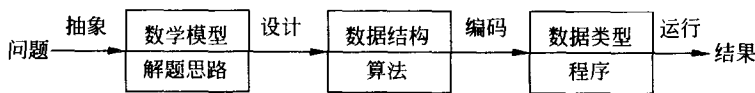


图 1-1 计算机解决问题的一般过程

由于当时所涉及的运算对象是简单的整型、实型或布尔型数据,所以程序设计者的主要精力集中于程序设计的技巧上,而无需重视数据结构。随着计算机应用领域的扩大和软、硬件的发展,非数值计算问题显得越来越重要。据统计,当今处理非数值计算性问题占用了 90% 以上的机器时间。这类问题涉及的数据结构更为复杂,数据元素之间的相互关系一般无法用数学方程式描述。因此,解决这类问题的关键不再是数学分析和计算方法,而是要设计出合适的数据结构。下面所列举的实例就是属于这一类的具体问题。

**例 1.1 学生信息检索系统。**当我们需要查找某个学生的有关情况,或者想查询某个专业或年级的学生的有关情况时,只要建立相关的数据结构,按照某种算法编写相关程序,就可以实现计算机自动检索。由此,可以在学生信息检索系统中建立一个按学号顺序排列的学生信息表,以及分别按姓名、专业、年级顺序排列的索引表,如图 1-2 所示。由这 4 个表构成的文件便是学生信息检索的数学模型,计算机的主要操作便是按照某个特定要求(如给定姓名)对学生信息文件进行查询。

诸如此类的还有电话自动查号系统、考试查分系统、仓库库存管理系统等。在这类文档管理的数学模型中,计算机处理的对象之间通常存在着一种简单的线性关系,因此,这类数学模型称为线性数据结构。

**例 1.2 教学计划编排问题。**一个教学计划包含许多课程,在这些课程中,有些课程必须按规定的先后次序进行学习,有些则没有次序要求。即有些课程之间有先修和后续的关系,有些课程可以任意安排次序。这种各个课程之间的次序关系可用称之为图的数据结构来表示,如图 1-3 所示。有向图中的每一个顶点表示一门课程,如果从顶点  $v_i$  到  $v_j$  之间存在有向边  $\langle v_i, v_j \rangle$ ,则表示课程  $i$  必须先于课程  $j$  进行。

由以上两个例子可见,描述这类非数值计算问题的数学模型不再是数学方程,而是诸如表、树、图之类的数据结构。因此,可以说数据结构课程主要是研究非数值计算的程序设计问题中所出现的计算机操作对象,以及它们之间的关系和操作的学科。

学号	姓名	性别	专业	年级
980001	吴承志	男	计算机科学与技术	98级
980002	李淑芳	女	信息与计算科学	98级
990301	刘丽	女	数学与应用数学	99级
990302	张会友	男	信息与计算科学	99级
990303	石宝国	男	计算机科学与技术	99级
000801	何文颖	女	计算机科学与技术	2000级
000802	赵胜利	男	数学与应用数学	2000级
000803	崔文靖	男	信息与计算科学	2000级
010601	刘丽	女	计算机科学与技术	2001级
010602	魏永鸣	男	数学与应用数学	2001级

(a) 学生信息表

崔文靖	8
何文颖	6
李淑芳	2
刘丽	3,9
石宝国	5
魏永鸣	10
吴承志	1
赵胜利	7
张会有	4

(b) 姓名索引表

计算机科学与技术	1,5,6,9
信息与计算科学	2,4,8
数学与应用数学	3,7,10

(c) 专业索引表

2000级	6,7,8
2001级	9,10
98级	1,2,3
99级	4,5

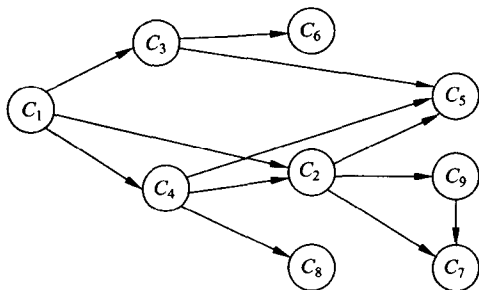
(d) 年级索引表

图 1-2 学生信息查询系统中的数据结构

学习数据结构的目的是了解计算机处理对象的特性,将实际问题中所涉及的处理对象在计算机中表示出来,并对它们进行处理。与此同时,通过算法训练提高学生的思维能力,通过程序设计的技能训练促进学生的综合应用能力和专业素质的提高。

课程编号	课程名称	先修课程
$C_1$	计算机导论	无
$C_2$	数据结构	$C_1, C_4$
$C_3$	汇编语言	$C_1$
$C_4$	C 程序设计语言	$C_1$
$C_5$	计算机图形学	$C_2, C_3, C_4$
$C_6$	接口技术	$C_3$
$C_7$	数据库原理	$C_2, C_9$
$C_8$	编译原理	$C_4$
$C_9$	操作系统	$C_2$

(a) 计算机专业的课程设置



(b) 课程之间优先关系的有向图

图 1-3 教学计划编排问题的数据结构

### 1.1.2 有关概念和术语

在系统地学习数据结构知识之前,先对一些基本概念和术语赋予确切的定义。

数据(data)是信息的载体,它能够被计算机识别、存储和加工处理。它是计算机程序加工的原料,应用程序可以处理各种各样的数据。它可以是数值数据,也可以是非数值数据。数值数据包括整数、实数或复数,主要用于工程计算、科学计算和商务处理等;非数值数据包括字符、文字、图形、图像、语音等。

数据元素(data element)是数据的基本单位,在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素可由若干个数据项组成。在不同的条件下,数据元素又可称为元素、结点、顶点、记录等。例如,学生信息检索系统中学生信息表中的一个记录,教学

计划编排问题中的一个顶点等,都被称为一个数据元素。

数据项(data item)是不可分割的、含有独立意义的、最小数据单位,有时也称为字段(field)或域。例如,学籍管理系统中学生信息表的每一个数据元素就是一个学生记录,它包括学生的学号、姓名、性别、籍贯、出生年月、成绩等数据项。这些数据项可以分为两种,一种是初等项,如学生的性别、籍贯等,这些数据项是在数据处理时不能再分割的最小单位;另一种是组合项,如学生的成绩,它可以再划分为数学、物理、化学等更小的项。通常,在解决实际应用问题时把每个学生记录当做一个基本单位进行访问和处理。

数据结构(data structure)是指相互之间存在一种或多种关系的数据元素的集合。在任何问题中,数据元素之间都不是孤立的,都会存在着这样或那样的关系,这种数据元素之间的关系称为结构。根据数据元素之间关系的不同特性,通常有以下4类基本的结构(数据的逻辑结构)。

#### (1) 集合结构

在集合结构中,数据元素间的关系是“属于同一个集合”。数据元素之间除了同属一个集合外,不存在其他关系。集合是元素关系中极为松散的一种结构。

#### (2) 线性结构

线性结构的数据元素之间存在一对一的关系。

#### (3) 树型结构

树型结构的数据元素之间存在一对多的关系。

#### (4) 图状结构

图状结构的数据元素之间存在多对多的关系。图状结构也称为网状结构。图1-4为上述4类基本结构的示意图。

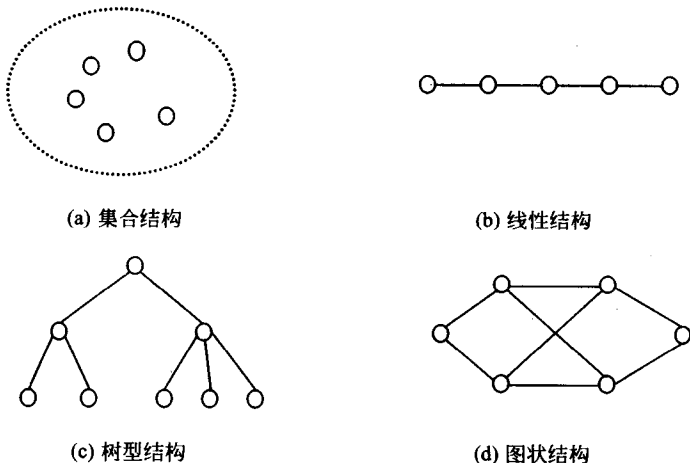


图 1-4 4类基本结构示意图

由于集合是数据元素之间关系极为松散的一种结构,因此也可用其他结构来表示它。

由数据结构的观念中可以知道,一个数据结构有两个要素,一个是数据元素的集合,另一个是关系的集合。在形式上,数据结构通常可以采用一个二元组来表示。

数据结构的定义形式为:数据结构是一个二元组

$$\text{Data\_Structure} = (D, R)$$

其中, $D$ 是数据元素的有限集, $R$ 是 $D$ 上关系的有限集。

数据结构包括数据的逻辑结构和数据的物理结构。数据的逻辑结构可以看做是从具体问题抽象出来的数学模型,它与数据的存储无关。研究数据结构的目的是为了在计算机中实现对它的操作,为此还需要研究如何在计算机中表示一个数据结构。数据结构在计算机中的表示(又称为映像)称为数据的物理结构,或存储结构。它所研究的是数据结构在计算机中的实现方法,包括数据结构中数据元素的表示和数据元素间关系的表示。

数据的存储结构可以采用顺序存储或链式存储方法。

顺序存储方法是把逻辑上相邻的元素存储在物理位置相邻的存储单元中,由此得到的存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法,通常借助于程序设计语言中的数组来实现。

链式存储方法对逻辑上相邻的元素不要求其物理位置相邻,元素间的逻辑关系通过附设的指针字段来表示,由此得到的存储表示称为链式存储结构,链式存储结构通常借助于程序设计语言中的指针类型来实现。

除了通常采用的顺序存储方法和链式存储方法外,有时为了查找方便,还采用索引存储方法和散列存储方法。

数据类型(data type)是一个与数据结构密切相关的概念。在高级程序设计语言中,用于限制变量取值范围和可能进行的操作的总和称为数据类型。数据类型可分为两类,一类是非结构的原子类型,如C语言中的基本类型(整型、实型、字符型等),指针类型和空类型;另一类是结构类型,它可以由多个结构类型组成,也可以分解。结构类型的成分中可以是非结构的,也可以是结构的。例如,数组的值由若干分量组成,每个分量可以是整数,也可以是数组等结构类型。

本书在讨论各种数据结构时,针对其逻辑结构和具体的存储结构给出相应的数据类型,进一步在确定的数据类型上实现各种操作。

算法(algorithm)是解决特定问题的一种方法或步骤的描述。

数据结构作为一门独立的课程起始于1968年,但在此之前与其有关的内容已散见于编译原理及操作系统中。20世纪60年代中期,美国的一些大学开始设立相关课程,但当时的课程名称并不叫数据结构。1968年,唐·欧·克努特教授开创了数据结构的最初体系,他所著的《计算机程序设计技巧》第1卷《基本算法》是第一本较系统地阐述数据的逻辑结构和存储结构及其操作的著作。从20世纪60年代末到70年代初,出现了大型程

序,软件也相对独立,结构程序设计成为程序设计方法学的主要内容。人们越来越重视数据结构,认为程序设计的实质是确定数据的结构,再加上设计一个好的算法,也就是人们所说的“程序=数据结构+算法”。

## 1.2 算法描述

算法与数据结构的关系紧密,在算法设计时,先要确定相应的数据结构,而在讨论某一种数据结构时也必然会涉及相应的算法。下面从算法特性、算法描述、算法性能分析与度量等三个方面对算法进行介绍。

### 1.2.1 算法特性

算法是对特定问题求解步骤的一种描述,是指令的有限序列。其中,每一条指令表示一个或多个操作。一个算法应该具有下列特性。

- 1) 有穷性。一个算法必须在有穷步之后结束,即必须在有限时间内完成。
- 2) 确定性。算法的每一步必须有确切的定义,无二义性,且在任何条件下算法只有惟一一条执行路径,即对于相同的输入只能得出相同的输出。
- 3) 可行性。算法中的每一步都可以通过已经实现的基本运算的有限次执行得以实现。
- 4) 输入。一个算法具有零个或多个输入,这些输入取自特定的数据对象集合。
- 5) 输出。一个算法具有一个或多个输出,这些输出与输入之间存在某种特定的关系。

算法的含义与程序十分相似,但又有区别。一个程序不一定满足有穷性。例如,对于操作系统,只要整个系统不遭到破坏,就永远不会停止工作,即使没有作业需要处理,它仍处于动态等待中。因此,操作系统不是一个算法。另一方面,程序中的指令必须是计算机可执行的,而算法中的指令则无此限制。算法代表了对问题的解,而程序则是算法在计算机上特定的实现。一个算法若用程序设计语言来描述,它就是一个程序。

算法与数据结构是相辅相成的。解决某一特定类型问题的算法可以选定不同的数据结构,而选择恰当与否将直接影响算法的效率。反之,一种数据结构的优劣由各种算法的执行来体现。

要设计一个好的算法通常应考虑以下要求。

- 1) 正确。算法的执行结果应当满足预先规定的功能和性能要求。
- 2) 可读。一个算法应当思路清晰、层次分明、简单明了、易读易懂。
- 3) 健壮。当输入不合法数据时,应能做适当处理,不至于引起严重后果。
- 4) 高效。能够有效地使用存储空间和具有较高的时间效率。



一般来说,数据结构的基本操作主要有以下几种。

- 1) 查找。寻找满足特定条件的数据元素所在的位置。
- 2) 读取。读出指定位置上数据元素的内容。
- 3) 插入。在指定位置上添加新的数据元素。
- 4) 删除。删除指定位置上对应的数据元素。
- 5) 更新。修改某个数据元素的值。

### 1.2.2 算法描述

算法的描述方法很多,根据描述算法语言的不同,可将算法描述分为以下4种。

#### (1) 框图算法描述

框图算法描述(程序流程图、N-S图等)在算法研究的早期曾经流行过。它的优点是直观、易懂,但用来描述比较复杂的算法就显得不够方便,也不够清晰简洁。

#### (2) 非形式算法描述

非形式算法描述用人类自然语言(如中文,英文等),同时还使用一些程序设计语言中的语句来描述算法,这称为非形式算法描述。

#### (3) 伪语言算法描述

如类C语言算法描述。这种算法不能直接在计算机上运行,但专业设计人员经常使用类C语言来描述算法,它容易编写、阅读和统一格式。

#### (4) 高级语言编写的程序或函数

这是可以在计算机上运行并获得结果的算法,使给定的问题能够在有限的时间内求解。通常,这种算法也称为程序。

下面以求两个整数  $m, n (m \geq n)$  的最大公因子为例,考察不同算法的描述方法。

(1) 该问题的框图描述(程序流程图)如图 1-5 所示。

#### (2) 非形式算法描述

- ① 求余数。以  $n$  除  $m$ , 并令  $r$  为余数 ( $0 \leq r < n$ )。
- ② 余数是零否。若  $r=0$  则结束算法,  $n$  就是最大公因子。

③ 替换并返回步骤①。若  $r \neq 0$  则  $m \leftarrow n, n \leftarrow r$  返回步骤①。

#### (3) C语言函数描述

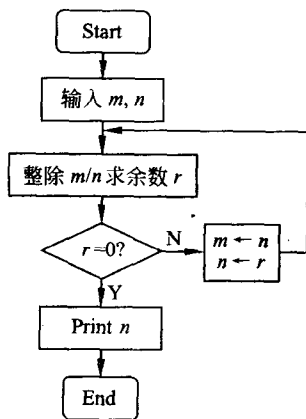


图 1-5 程序流程图