普通高等教育“十二五”规划教材

21世纪高等学校精品教材

【本书配有教学课件】

SHUJUJIEGOULILUNYUSHIJIAN

数据结构理论与实践

杨永斌⊙主编

天津科学技术出版社

提倡低碳生活

数据结构理论与实践

主　编　杨永斌

副主编　（以姓氏笔画为序，排名不分先后）

杨友斌　许海成　顾东虎

晏　轲　谢景伟　雷　鸿

戴小鹏

编　委　（以姓氏笔画为序，排名不分先后）

于春霞　石　冰　叶星火

史　永　杨玉光　李红育

宋蓓蓓　周绍景　贾家新

葛颖增

天津科学技术出版社

图书在版编目（CIP）数据

数据结构理论与实践/杨永斌主编. —天津：
天津科学技术出版社，2011.5

ISBN 978-7-5308-6346-6

Ⅰ.①数… Ⅱ.①杨… Ⅲ.①数据结构 Ⅳ.①TP311.12

中国版本图书馆 CIP 数据核字（2011）第 085939 号

责任编辑：吴文博
责任印制：兰　毅

天津科学技术出版社出版
出版人：蔡颢
天津市西康路 35 号　邮编 300051
电话（022）23332398（事业部）　23332697（发行）
网址：www.tjkjcbs.com.cn
新华书店经销
北京佳艺丰印刷有限公司印刷

开本 787×1092　1/16　印张 20　字数 420 000
2011 年 6 月第 1 版第 1 次印刷
定价：39.00 元

前　言

数据结构是信息与计算科学专业的专业课程，主要介绍用计算机解决一系列问题特别是非数值信息处理问题时所用的各种组织数据的方法、存储数据结构的方法以及在各种结构上执行操作的算法。通过对本书的能让学生掌握各种数据结构的特点、存储表示、运算方法以及在计算机科学中最基本的应用，培养、训练学生选用合适的数据结构和编写质量高、风格好的应用程序的能力，并为后续课程的学习打下良好的理论基础和实践基础。

本书共分8章，第1章绪论，主要介绍数据、数据结构和算法等基本概念。第2章至第6章分别讨论线性表、线、队列等基本类型的数据结构。第7章和第8章讨论查找和排序。涵盖了“数据结构”课程学习所需实践的各个方面和“数据结构”课程的主要内容，可作为高等院校学生学习“数据结构”课程的理论与实践教材。

本书以C语言为程序设计语言，采用系列式的叙述方式，引导读者循序渐进地掌握数组、链接表、栈和队列、树与森林、图和堆等不同的数据结构，并系统地介绍了查找和排序的各种方式。对每一种数据结构，除了详细阐述其基本概念和具体实现外，都尽可能地对每种操作给出C语言的算法描述；对查找和排序的各种算法，还着重在时间上做出定量或定性的分析比较。

本书提供了大量的源程序，对于从事计算机应用及开发的技术人员、从事数据结构和程序设计教学或学习的教师和学生具有很好的参考和指导作用。

本书在编写过程中，得到全国几十所高校的领导和教师的大力支持与帮助。借此机会向他们表示感谢！教材编写中，力求做到尽美，但由于水平有限，难免存在不妥，疏漏之处，恳请同行专家、读者不吝赐教。

编　者

2011年3月

目 录

第1章 绪论

[学习目标]

本章作为全书的导引，介绍了组织数据和设计算法的相关知识，引入了数据、数据元素、数据结构、逻辑结构、存储结构、数据运算、算法描述和算法评价等基本概念。

[基本要求]

通过学习本章，掌握以下内容：

①理解和熟悉数据结构的定义及其实现方法；

②掌握算法的评价规则；

③算法时间复杂度、空间复杂度的概念和评价的表示方法。

1.1 数据结构的重要性

电子计算机是20世纪科学技术最卓越的成就之一。自1946年世界上第一台电子计算机问世以来，计算机产业和应用的发展远远超出了人们对它的预料。在计算机发展的初期，人们使用计算机主要是处理数值的计算问题，程序设计人员也把主要精力集中在程序设计的技巧上。随着计算机技术的飞速发展，计算机的应用范围不断扩大，已经不再局限于单纯的数值计算，更多地被应用于控制、管理及数据处理等非数值计算的处理工作。加工处理的信息也由简单的数值发展到字符、图像、声音等具有复杂结构的数据。

非数值型问题在人们的日常生活中是非常多的，也需要我们使用计算机来处理这些非数值问题。例如：在城市交通运输中，从A点到B点有很多条道路，每条道路的长度不同、拥挤程度不同，我们要选择一条最快的线路到达目的地，该如何选择？再比如，图书馆有成千上万条图书资料，我们该如何进行管理，才能快速查找到需要的资料？北京市有上千万的人口，我们应该怎样保存这些人口的资料信息，才能快速查找到需要查找的人？像这样的问题都是典型的非数值问题。

要用计算机处理这些非数值问题，就为我们提出了一个课题：如何在计算机内部描述这些非数值问题，采用什么样的算法可以快速、有效地完成问题的求解。用计算机解决任何问题都离不开程序设计。为了编制好的程序，必须分析程序处理的数据特性及数据之间的关系，这就是数据结构这门学科形成和发展的背景。

进一步地，对于不同的处理对象，要想设计出高质量的程序，就必须研究如何组织数据和处理数据，根据问题的要求及数据元素之间的特性，确定相应的存储结构和算法，这些都是数据结构研究的内容。

用计算机解决一个具体问题时，大致需要经过下列3个步骤：

①从具体问题抽象出一个适当的数学模型；

②设计一种解此数学模型的算法；

③编制程序，进行测试、调整，直至得到最终解答。

程序设计语言研究的范畴在第③步，而数据结构重点研究的属于数据模型如何形成算法的范畴。可以通过下面的例子来认识数据结构。

电话是联络通讯必不可少的工具。如何用计算机来实现高效自动查询电话号码呢？要求对于给定的任意姓名，若该人有电话号码，则迅速给出电话号码；否则，给出查找不到该人电话号码的信息。

对于这样的问题，可以按照客户向电信局申请电话号码的先后次序建立电话号码表，存储到计算机中。在这种情况下，由于电话号码表是没有任何规律的，查找时，只能从第一个号码开始，逐一进行，这样逐一按照顺序进行查找的效率是非常低的。为了提高查询的效率，可以根据每个用户姓名的第一个拼音字母，按照26个英文字母的顺序进行排列，这样，根据姓名的第一个字母，就可以迅速地进行查找，从而极大地减少了查找所需要的时间。进一步地，我们可以按照用户的中文姓名的汉语拼音顺序进行排序，这样就可以进一步提高查询效率。

在上述例子中，人们感兴趣的是如何提高查找效率。为了解决这个问题，就必须了解待处理对象之间的关系，以及如何存储和表示这些数据。在电话号码查询的例子中，每一个电话号码就是一个要处理的数据对象，也称为数据元素，在数据结构中，为了抽象地表示不同的数据元素，为了研究具有相同性质的数据元素的共同特点和操作，又将数据元素称为数据结点，简称为结点。电话号码经过处理，按照拼音排好了顺序，每个电话号码之间的先后次序就是数据元素之间的关系。数据结构就是研究这类非数值处理的程序设计问题的。

据统计，当今处理非数值计算性问题占用了85%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加以描述。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，才能有效地解决问题。下面例子就是属于这一类的具体问题。

在图书馆信息检索系统中，当根据书名查找某本书的有关情况的时候，或者根据作者或某家出版社查找有关书籍的时候，或根据书刊号查找作者和出版社等有关情况的时候，只要建立了相关的数据结构，按照某种算法编写了相关程序，就可以实现计算机自动检索。由此，可以在图书馆信息检索系统中建立一张按照书刊号顺序排列的图书信息表和分别按照作者、书名、出版社顺序排列的索引表，如图1．1所示。由这四张表构成的文件便是图书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（如给定书名），对图书馆藏书信息文件进行查询。

诸如此类的还有学生信息查询系统、商场商品管理系统、仓库物资管理系统等。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一种简单的线性关系，这类数学模型可称为线性的数据结构。

书刊号	书名	作者	出版社	书架编号
7－302－02368－3	数据结构	严蔚敏	清华大学出版社	p302－419－2
5－301－00128－1	线性代数	李志中	复旦大学出版社	k203－048－1
7－5308－6346－6	数据结构	杨永斌	天津科学技术出版社	p302－428－5
5－303－01654－3	线性代数	赵坚	清华大学出版社	k205－046－2
6－506－04314－2	离散数学	左孝凌	复旦大学出版社	t－304－056－2
7－406－03416－1	数据结构	李有亮	上海教育出版社	p302－433－4
7－406－01256－2	计算机组成	顾浩	清华大学出版社	w306－224－2
7－5308－6294－0	离散数学	高栓虎	天津科学技术出版社	t－306－433－5
7－408－056118－2	计算机组成	顾浩	上海教育出版社	w－305－036－6

图书信息表（a）

数据结构	1，3，6
线性代数	2，4
离散数学	5，8
计算机组成	7，9

书名索引表（b）

杨永斌	3
顾浩	7，9
李志中	2
李有亮	6
严蔚敏	1
高检虎	8
左孝凌	5
赵坚	4

作者索引表（c）

清华大学出版社	1，4，7
复旦大学出版社	2，5
天津科学技术出版社	3，8
上海教育出版社	6，9

出版社索引表（d）

图1.1　图书馆信息检索

在八皇后问题中，处理过程不是根据某种确定的计算法则，而是利用试探和回溯的探索技术求解。为了求得合理布局，在计算机中要存储布局的当前状态。从最初的布局状态开始，一步一步地进行试探，每试探一步，形成一种新的状态，整个试探过程形成了一棵隐含的状态树。如图1.2所示（为了描述方便，将八皇后问题简化为四皇后问题）。回溯法求解过程实质上就是一个遍历状态树的过程。在这个问题中所出现的树也是一种数据结构，它可以应用在许多非数值计算的问题中。

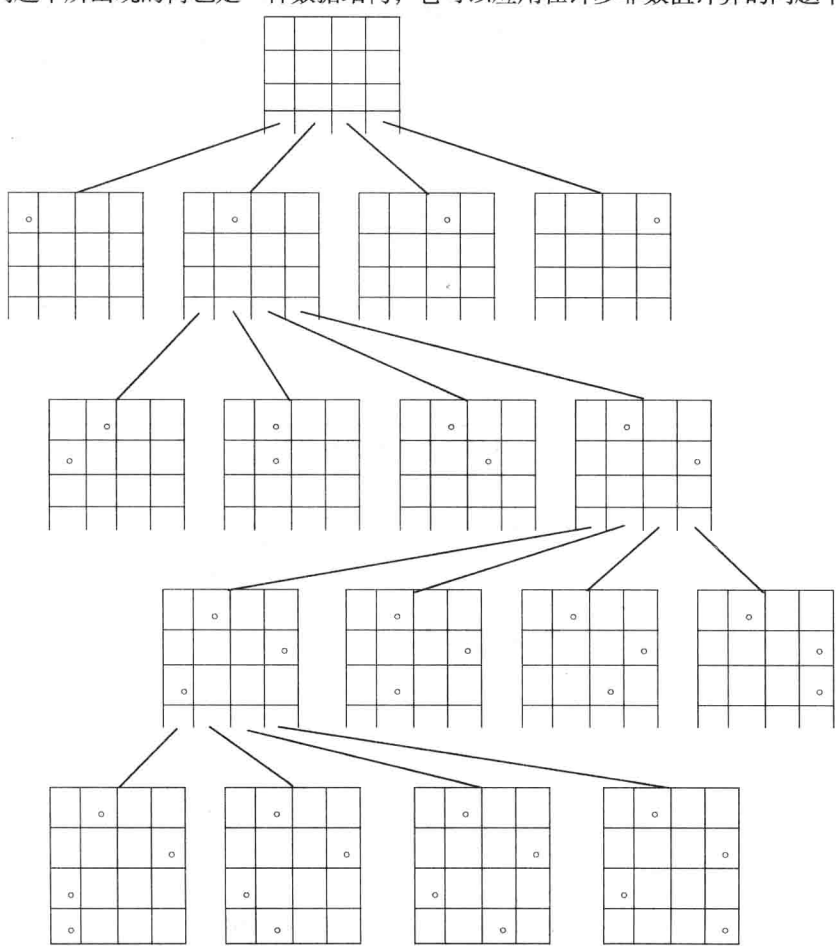


图1.2　四皇后问题中隐含的状态树

在教学计划编排问题中，一个教学计划包含许多课程，在教学计划包含的许多课程之间，

有些必须按照规定的先后次序进行，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。这种各门课程之间的次序关系可用一个被称做图的数据结构来表示，如图1．3所示。有向图中的每个顶点表示一门课程，若从顶点 v_i 到 v_j 之间存在有向边 $<v_i, V_j>$，则表示课程 i 必须先于课程 j 进行。

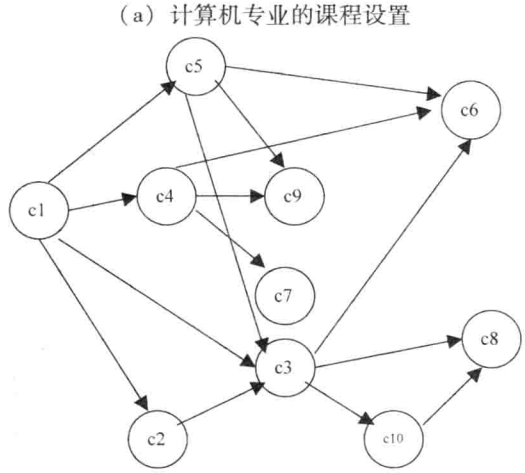
由以上三个例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如线性表、树、图之类的数据结构。因此可以说，数据结构课程主要是研究非数值计算的程序设计问题中出现的计算机操作对象以及它们之间的关系和操作的学科。

学习数据结构课程的目的是为了了解计算机处理对象的特性，将实际问题中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，通过算法训练来提高学生的思维能力，通过程序设计的技能训练来促进学生的综合应用能力和专业素质的提高。

数据结构课程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各种类型的数据结构。因此，要想更好地运用计算机来解决实际问题，要想有效地使用计算机、充分发挥计算机的性能，需要学习和掌握好数据结构构的有关知识。学好数据结构这门课程，对于学习计算机专业的其他课程，如操作系统、数据库应用技术、软件工程等课程都是十分有益的。

课程编号	课程名称	先修课程
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_5
C_4	汇编语言	C_1
C_5	C 程序设计语言	C_1
C_6	计算机图形学	C_3，C_4，C_5
C_7	接口技术	C_4
C_8	数据库原理	C_3，C_{10}
C_9	编译原理	C_4，C_5
C_{10}	操作系统	C_3

（a）计算机专业的课程设置



（b）表示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

1.2　基本概念和术语

数据结构在计算机科学中是一门综合性的专业基础课。数据结构的研究不仅涉及计算机的硬件（特别是编码理论、存储装置和存取方法等）的研究范围，而且和计算机软件的研究有着更密切的关系，无论是编译程序，还是操作系统，都涉及数据元素在存储器中的分配问题。在研究信息检索时，也必须考虑如何组织数据，以使查找和存取数据元素更为方便。因此可以认为，数据结构是相关数学、计算机硬件和计算机软件的一门计算机专业核心课程，其关系如图1.4所示。在计算机科学中，数据结构不仅是一般程序设计（特别是非数值计算的程序设计）的基础，而且是设计和实现编译程序、操作系统、数据库系统及其他系统程序和大型应用程序的重要基础，被广泛地应用于信息学、系统工程等各种领域。Pascal创始人Niklaus Wirth提出：

Algorithm + Data Structures = Programs

{算法 + 数据结构 = 程序设计}

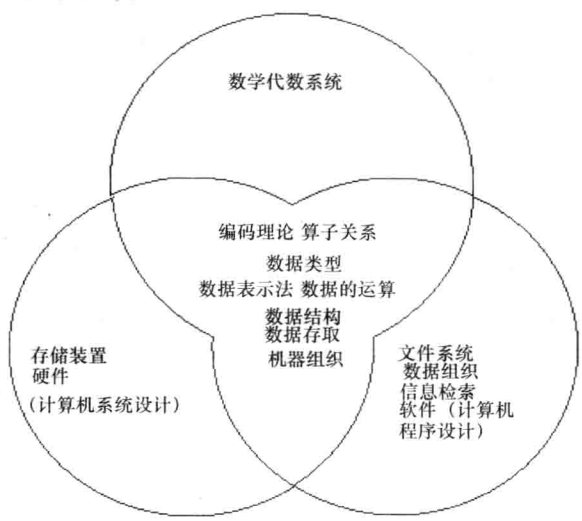


图1.4　数据结构和其他课程之间的关系

1.2.1　基本概念

（1）数据（Data）

数据是对客观事物的符号表示，在计算机科学中，是指所有能输入到计算机中并被计算机程序处理的符号的总称。

（2）数据元素（Data Element）

数据元素是数据的基本单位，在计算机程序中，通常作为一个整体进行考虑和处理。有时，一个数据元素可由若干个数据项（Data Item）组成，数据项是数据的不可分割的最小单位。

（3）数据对象（Data Object）

数据对象是性质相同的数据元素的集合，是数据的一个子集。例如，整数数据对象是集合N = {0，±1，±2，…}，字母字符数据对象是集合C = {'A','B'，…,'Z'}。

（4）数据结构（Data Structure）

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构的形式定义为：

数据结构是一个二元组，Data_ Structure = （D，S），其中：D 是数据元素的有限集，S 是 D 上关系的有限集。

在计算机科学中，复数可用如下定义：复数是一种数据结构，Complex = （C，R），其中：C 是含两个实数的集合 $\{c_1, c_2\}$；R = {P}，而 P 是定义在集合 C 上的一种关系 $\{<c_1, c_2>\}$，其中有序偶 $<c_1, c_2>$ 表示 c_1 是复数的实部，c_2 是复数的虚部。

假设需要编制一个学校科学研究的事务管理的程序，管理学校科学研究课题小组的各项事务，则首先要为程序的操作对象——课题小组——设计一个数据结构。假设每个小组由一位教师、一至三名研究生及一至六名本科生组成，小组成员之间的关系是：教师指导研究生，而由每位研究生指导一至两名本科生。则可以如下定义数据结构：

$$Group = (P, R)$$

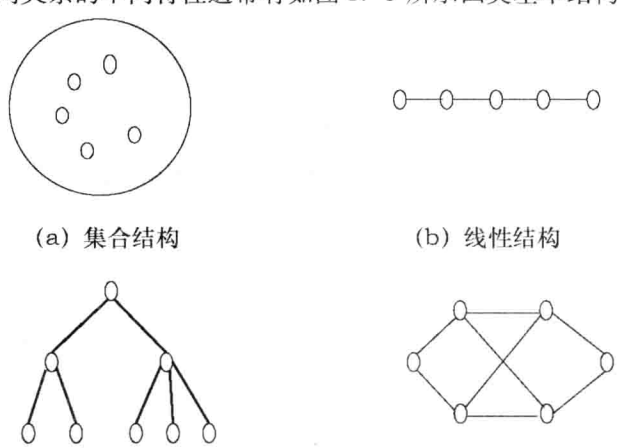
其中，$P = \{T, G_1, \cdots, G_n, S_{11}, \cdots, S_{nm}\}$，$1 <= n <= 3$，$1 <= m <= 2$，$R = \{R_1, R_2\}$

$R1 = \{<T, G_i> \mid 1 <= i <= n, 1 <= n <= 3\}$

$R2 = \{<Gi, S_{ij}> \mid 1 <= i <= n, 1 <= j <= m, 1 <= n <= 3, 1 <= m <= 2\}$

数据结构一般包含数据的逻辑结构、数据的存储结构和数据的运算三方面内容。数据的逻辑结构是指数据元素之间的逻辑关系，与数据在计算机内部如何存储无关，数据的逻辑结构独立于计算机。例如，在城市交通中，两个地点之间就存在一种逻辑关系，两个地点之间的逻辑关系分为三种。第一种，两个地点之间有公共汽车可以直达；第二种，两个地点之间没有公共汽车可以直达，但可以通过中途换乘其他公共汽车而到达；第三种逻辑关系是两个地点之间没有公共汽车可以达到。又如在电话号码本中，电话号码如何进行分类，按照什么顺序进行排列等，都是数据之间的逻辑关系。

为了进一步研究数据的逻辑结构，可以将数据的逻辑结构分为线性结构和非线性结构两大类。根据数据元素之间关系的不同特性通常有如图 1．5 所示四类基本结构。



（a）集合结构　　（b）线性结构

（c）树型结构　　（d）图形结构

图 1．5　四类基本结构的示意图

线性结构包括线性表、栈和队列等。其主要特征为各个数据之间有明确的、唯一的“先后”顺序。在现实生活中，具有线性结构的实例非常多，例如我们在日常生活中的排队购物，队列中的每个人都有一个明确的先后次序关系。

非线性结构包括树、图和集合型结构。

树型结构的主要特征是结点之间存在一种层次关系，每一个结点对应下一层的多个结点，也就是说，数据元素之间的关系是“一对多”的关系。例如一所大学下面有几个系，每个系下面有许多班，每个班下面有许多学生，即大学—系—班—学生，每一层之间都是一种一对多的关系，这就是一个典型的树型结构。

而在图型结构中，任何两个结点之间都可能存在联系，数据元素之间存在多对多的关系。典型的图型结构就是城市交通。若城市中有单行线路，则从城市中的一个地点 A 出发，可以到达 N 个不同的地方，从城市的 M 个不同的地方出发又可以到达地点 A。城市交通就是一个典型的“多对多”的图型结构。

集合型数据结构中的数据元素之间除了“同属于一个集合”的关系外，别无其他关系。数据的存储结构是指数据元素在计算机存储设备中的存储方式，既可以用顺序存储方式，也可以用链式存储方式。顺序存储方式的特点是借助元素在存储器中的相邻位置来表示数据元素之间的逻辑关系。非顺序存储方式的特点是借助指示元素存储地址的指针（Pointer）表示数据元素之间的逻辑关系。

在城市交通的例子中，要研究如何在计算机中表示一个地点，如何在计算机中表示两个地点之间存在一条公共汽车线路，该线路有多长等。又如对于电话号码资料信息在计算机中如何保存，如何表示资料的分类，如何表示排好顺序的电话号码之间的先后次序关系。

数据的运算主要包括插入、删除、检索和排序等与问题相关的处理。数据结构通常具有下列一些基本操作。

①插入：在数据结构的指定位置上添加一个新结点。

②删除：删去数据结构中指定位置的结点。

③更新：修改数据结构中某个结点的值。

④查找：在数据结构中寻找满足指定条件的结点及其位置。

⑤排序：按照指定的顺序，使结点重新排列。

例如，在城市交通问题中，要设计求两点之间最短路线的算法，要能够判断从城市的任一点出发乘坐公共汽车是否可以达到城市的任何地方。在电话号码问题中，要设计如何插入一个新的电话号码信息，如何删除一个作废的电话号码信息，如何对电话号码进行快速整理排序，如何高效快速查找资料等算法。

1．2．2　数据类型

按照“值”的不同特性，高级程序语言中的数据类型（Data Type）可分为两类：一类是非结构的原子类型。原子类型的值是不可分解的。例如，C 语言中的基本类型（整型、实型、字符型和枚举类型）、指针类型和空类型。另一类是结构类型。结构类型的值是由若干成分按照某种结构组成的，因此是可以分解的，并且它的成分既可以是非结构的，也可以是结构的。

抽象数据类型（Abstract Data Type，简称 ADT）是指一个数学模型和定义在该模型上的一组操作，面向逻辑层次。抽象数据类型的定义仅仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关，即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。

采用介于伪码和 C 语言之间的类 C 语言作为描述工具，简要说明如下。

①预定义常量和类型格式如表 1．1 所示。

表 1.1 预定义常量和类型

函数结果	状态代码
#define TRUE	1
#define FALSE	0
#define OK	1
#define ERROR	0
#define INFEASIBLE	-1
#define OVERFLOW	-2

②数据结构的表示（存储结构）用类型定义（Typedef）描述。数据元素类型约定为 ElemType，由用户在使用该数据类型时自行定义。

③基本操作的算法都用以下形式的函数描述：

```
函数类型    函数名（函数参数表）
            {
            语句序列
            }
```

除了函数的参数需要说明类型外，算法中使用的辅助变量可以不作变量说明，必要时，对其作用给予注释。一般而言，a，b，c，d，e 等用做数据元素名，i，k，l，m，n 等用做整型变量名，p，q 等用做指针变量名。当函数返回值为函数结果状态代码时，函数定义为 Status 类型。为了便于算法描述，除了值调用方式外，增添了C++语言的引用调用的参数传递方式。在形参表中，以 & 打头的参数即为引用参数。

④赋值语句格式如表 1.2 所示。

表 1.2 赋值语句格式表

语句	格式
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 k = 表达式;
成组赋值	（变量名 1，…，变量名 k）=（表达式 1，…，表达式 k）; 结构名 = 结构名; 结构名 =（值 1，…，值 k）; 变量名［］= 表达式; 变量名［起始下标.. 终止下标］= 变量名［起始下标.. 终止下标］
交换赋值	变量名 <——> 变量名;
条件赋值	变量名 = 条件表达式? 表达式 T：表达式 F

⑤选择语句格式如表 1.3 所示。

表 1.3 选择语句格式表

语句	格式
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	switch（表达式）{ case 值 1：语句序列 1；break; … case 值 n：语句序列 n；break; default：语句序列 n + 1;}

⑥循环语句格式如表 1.4 所示。

表1.4　循环语句格式表

语句	格式
for	for（赋初值表达式序列；条件；修改表达式序列）语句；
while	while（条件）语句；
do-while	do ｛ 语句序列； ｝while（条件）；

⑦结束语句格式如表1.5所示。

表1.5　结束语句格式表

语句	格式
函数结束语句	return 表达式 return；
case 结束语句	break；
异常结束语句	exit（异常代码）；

⑧输入和输出语句格式如表1.6所示。

表1.6　输入和输出语句格式表

语句	格式
输入语句	scanf（［格式串］，变量1，…，变量n）； 或 cin＞＞变量1＞＞…＞＞变量n；
输出语句	printf（［格式串］，表达式1，…，表达式n）； 或 cout＜＜表达式1＜＜…＜＜表达式n；

通常省略格式串。

⑨注释。

单行注释：//文字序列。

⑩基本函数格式如表1.7所示。

表1.7　基本函数格式表

语句	格式
求最大值	max（表达式1，…，表达式n）
求最小值	min（表达式1，…，表达式n）
求绝对值	abs（表达式）
求不足整数值	floor（表达式）
求进位整数值	ceil（表达式）
判定文件结束	eof（文件变量）或 eof
判定行结束	eoln（文件变量）或 eoln

⑪逻辑运算约定格式如表1.8所示。

表1.8　逻辑运算约定格式表

语句	格式
与运算 &&	对于A&&B，当A的值为0时，不再对B求值。
或运算‖	对于A‖B，当A的值为非0时，不再对B求值。