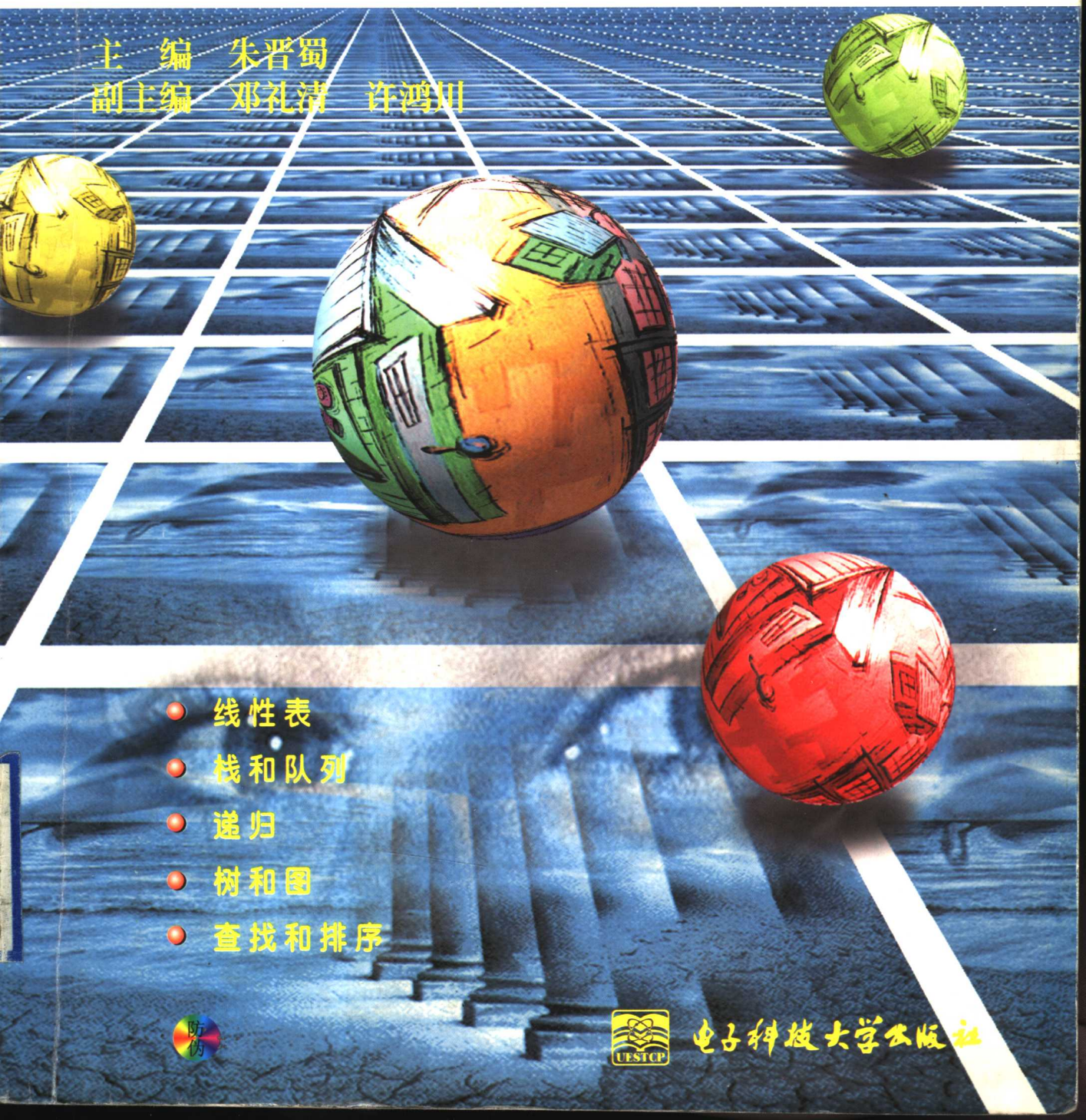
高等学校计算机系列教材

数据结构

主 编 朱晋蜀
副主编 邓礼清 许鸿川



- 线性表
- 栈和队列
- 递归
- 树和图
- 查找和排序

UESTCP 电子科技大学出版社

高等学校计算机系列教材

数 据 结 构

丛 书 主 编：刘甫迎
丛书副主编：朱晋蜀　党晋蓉　杨明广　廖亚平
丛 书 编 委：邓礼清　王道学　姜文国　倪继烈　刘枝盛　许鸿川
蒋正萍　宋国明　刘新民　刘　虹　张　京　陈　琳
岳德坤　李　琦　刘光会　饶　斌　蔡方凯
主　　编：朱晋蜀
副 主 编：邓礼清　许鸿川

电子科技大学出版社

内 容 简 介

本书系统地介绍了各种常用的数据结构，主要内容有：绪论、线性表、栈和队列、串、递归、树、图、查找、排序、文件等。本书内容丰富，概念叙述清楚，每章后附有习题，适合教学以及学生自学。本书注重对学生的应用能力和实践能力的培养。全书采用 C 语言作为数据结构和算法的描述语言，便于学生阅读算法和上机实践。

本书可作为高等院校计算机专业的教材，也可供从事计算机研究与应用开发工作的科技人员参考。

高等学校计算机系列教材

数 据 结 构

主 编 朱晋蜀

副主编 邓礼清 许鸿川

出 版：电子科技大学出版社(成都建设北路二段四号 邮编：610054)

责任编辑：朱 丹

发 行：新华书店

印 刷：西南冶金地质印刷厂

开 本：787×1092 1/16 印张 11.75 字数 286 千字

版 次：2000 年 1 月第一版

印 次：2000 年 3 月第一次印刷

书 号：ISBN 7—81065—325—3/TP・204

印 数：4001—8000 册

定 价： 13.80 元

序

诞生于本世纪中叶的计算机科学较之其他现代科学技术的发展更迅速，在世纪之交到来之际，它几乎可以称为“知识爆炸”了。21 世纪是知识经济和信息的时代，信息技术的发展水平、运用水平和教育水平已经成为衡量社会进步的重要标志。面对挑战与机遇并存的发展形势，世界范围内的多层次、多侧面的计算机教育热潮正在蓬勃掀起。

要使得计算机教育和学习水平跃上一个新台阶，首先要提高对计算机教学重要地位和计算机应用基本目标的认识。显然，计算机的广泛普及与应用，使人们传统的工作、学习、生活、乃至思维方式都发生了巨大变化。不会利用计算机进行读写，不会利用计算机进行思维、工作和学习，将成为下一世纪的“文盲”。另一方面，计算机技术与其他学科领域交叉融合，促进了学科发展与专业更新，引发了新兴交叉学科与技术不断涌现。人们若不能很好地使用计算机，将无法掌握最先进、最有效的研究与开发手段，直接影响到其所从事专业的发展。计算机基础如同数学和外语等一样，已经成为面向 21 世纪人才培养方案中必不可少的、最重要的基础之一，必须花大力气搞好计算机教学。

高等学校计算机教学分为非计算机专业的计算机基础教学与计算机专业教学。前者的目标是：使学生掌握计算机软、硬件技术的基本知识，培养学生在本专业与相关领域中的计算机应用开发能力，培养学生利用计算机分析问题、解决问题的意识，提高学生的计算机文化素质。后者的目标应是：使学生有较扎实、系统的计算机软、硬件技术知识，具有安装、调试并维护前、后端数据库管理系统(DBMS)和客户/服务器模式的计算机网络的能力，能开发研制基于上述网络模式的管理信息系统(MIS)和其他应用软件(如图形、多媒体软件等)；能进行 Internet 网上的开发和应用；能进行计算机一般故障的维修等。非计算机专业教学与计算机专业教学两者不能截然分开，往往后者又是前者深入、拓展后学习者要求的必然。人们希望有一套计算机教学丛书能满足此需求。

基于上述需求的呼唤和为了全面提高学生的计算机业务素质，我们编写了这套“高等学校计算机系列教材”。

本系列教材的特点是：

1. 这些书的作者是一些长期从事计算机教学和科研的教师，不少作者在以前都有大量计算机方面的著作出版。例如，本系列教材中《Visual FoxPro 实用教程》一书的作者，10 多年前回国后最早将“狐狸”软件介绍到祖国大陆，这一本书已是他的第九本 Fox 方面的著作了。《数据结构》一书的作者是全国高校大专计算机专业教学指导委员会的委员，这一本书已是他的第六本著作。本系列教材中《计算机应用基础》一书的作者是四川省普通高等学校非计算机专业等级考试委员会委员，本身就是四川省计算机等级考试大纲的起草者之一，并多次参加计算机等级考试的命题工作，他以前参加编写的有关计算机等级考试的书已获四川省优秀教材奖。坚实的作者基础是这套书质量的最根本的保证。

2. 本系列教材是面向 21 世纪的计算机教学的教材，其内容既体现了最新计算机科学发展的先进性(例如，《Visual FoxPro 实用教程》就是以 1998 年 8 月 26 日才推出的最新版

本 Visual FoxPro 6.0 为背景写的），又注意了其内容的基础性。

3. 本系列教材可以根据不同读者的需求进行课程体系的组合。计算机专业的读者可以按如下顺序学习：

《微积分与工程数学》，《计算机应用基础》，《计算机电路基础》，《C++语言程序设计》，《数据结构》，《Visual BASIC 教程》，《Visual FoxPro 实用教程》，《微机原理与接口技术》，《操作系统》，《计算机网络技术》，《微型计算机故障诊断与维护》，《Windows NT 教程》，《Internet（因特网）及其应用》，《Photoshop 与三维动画》。这里已将“面向对象的程序设计”、“多媒体技术”、“Windows 编程”、“软件工程”、“操作系统”、“计算机网络技术”以及“Web 页面制作”等内容融合到这套书的相应课程中了。本系列教材注意了以“必须和够用”为度，既注意了前后教材之间的衔接，又避免了内容的重复（例如，OLE 的内容在 VB 中是很重要的，但由于在《Visual FoxPro 实用教程》中对 Windows 平台的 OLE 已作了详细讲解，故在 VB 中便不再赘述它了）。

非计算机专业的读者可以将本系列教材的《计算机应用基础》、《Visual BASIC 教程》和《Visual FoxPro 实用教程》等作为国家教委提出的计算机基础教育“三个层次”（即第一层次为计算机文化基础，第二层次为包括计算机语言、结构化程序设计和面向对象程序设计的计算机技术基础，第三层次为包括计算机信息管理基础与多媒体应用基础等的计算机应用基础）的主干课程，其他教材可选学，各书中带 * 号的内容可以不学。

4. 本系列教材强调了实用性和实践性。各书都有教学大纲和实验指导书，便于教师的教学和读者的上机实践。

编写一套系列教材，是一个巨大的系统工程。这套书的作者们、电子科技大学出版社的领导们和编辑们，都为她的诞生付出了辛勤的劳动。她的成长，更离不开大家的扶持。

希望广大读者多提批评意见，以利这套系列教材今后的改进。

希望读者们能喜欢这套书。

编委会

1998 年 11 月 20 日

前　言

数据结构是计算机类专业最基础的课程之一，也是一个程序员或软件工程师的必修课。无法想象没有学过数据结构课程的人能够设计、编写出质量与效益兼优的程序来。

“算法+数据结构=程序”，在一个实际应用编程时，首先是要考虑如何存储数据，换句话说，在计算机内存中是以某种结构来存储需处理的数据。在已确定的“数据存储结构”的基础上再来考虑使用哪一种算法。一般来说，数据结构与算法之间有着密不可分的关系，使用不同的数据结构，选择的算法就有所不同。

数据结构研究的存储结构是线性表、树、图，存储的方式有顺序结构和链式结构。除此之外，数据结构课程还研究一些基本的、典型的算法，如排序和查找以及递归的编程方法。

对于大多数初学者来讲，最难的也许是数据的抽象。假设要编写一个求 1～100 和 $\sum_{i=1}^{100} i$ 的程序，一种简单的算法是使用自然数顺序相加的办法来实现。而另一种算法(高斯算法)却是用“50×101”来实现的。高斯算法的可贵之外就是对需处理的数据进行了加工，找到了一种最简便的数据(结构)表达形式。

该课程的前驱课程是 C 语言程序设计。在本书中，算法的实现是用 C 语言实现的。

本书由朱晋蜀担任主编，邓礼清、许鸿川担任副主编。朱晋蜀编著了第一章、第四章；邓礼清编著了第五章、第六章、第七章、第十章；许鸿川编著了第二章、第三章、第八章、第九章、附录 A；附录 B 由邓礼清、许鸿川合写；全书由朱晋蜀统稿。

由于作者水平有限，错误在所难免，敬请读者批评指正。

作者

1999 年 9 月

目　　录

第一章 绪 论

数据结构是设计和实现编译程序、操作系统、数据库系统以及其他的系统程序和应用程序的重要基础。数据结构是研究数据之间的各种关系及其算法的实现。本章将讨论数据结构和算法的描述的基本概念。

1.1 数据结构的基本概念

在本节中将对一些概念和术语给出确定的含义，这些概念和术语将在本书以后的章节中多次出现。

数据(Data)是人们利用文字符号、数字符号以及其他规定的符号对现实世界的事物及其活动所作的描述。人们把能够被计算机输入、存储、处理和输出的一切信息都称为数据。因此，一本书、一篇文章、一张图表等都是数据，图像、声音、动画等也属于数据的范畴。

数据元素(Data Element)是数据实体中相对独立的、不可分割的基本数据单位。例如，对于一个字符串来说，每个字符就是它的数据元素，对于一个文件来说，每个记录就是它的数据元素，一张名片由姓名、职务、电话和住址等信息组成，姓名、职务、电话和住址等是名片的数据元素。对于一个数组来说，每一个成分是它的数据元素。数据和数据元素是相对而言的，例如，对于一个记录来说，它相对于所在的文件是一个数据元素，而它相对于所含的数据项又被认为是数据。在本书中，对于数据和数据元素这两个术语的使用并不加以严格的区别。

数据记录(Data Record)是数据处理领域组织数据的基本单位。数据记录由数据项(Item)所组成，数据项是比数据记录更小的单位，一个记录包括一个或若干个固定的数据项。例如对于名片管理来说，每张名片是一条记录，描述一个人的信息，姓名、职务、电话等相当于数据项。

数据对象(Data Object)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{0，±1，±2，…}，字母字符数据对象是集合{'A'，'B'，'C'，…，'Z'}。

数据处理(Data Processing)是指对数据进行查找、插入、删除、合并、排序、统计、计算和输出等操作过程。对于名片管理的数据处理，就有名片的查找、新名片的插入、名片的排序、名片打印输出等。

数据结构(Data Structure)这个概念至今尚未有一个被一致公认的定义，不同的人在使用这个词时所表达的意思有所不同，本书简单地说成是指数据以及数据之间的联系。在任何问题中，数据元素都不是孤立存在的，它们之间存在着某种关系，数据元素之间的这种相互关系称为结构(Structure)。

为了更确切地描述数据结构，通常采用离散数学中的二元组表示：

$$B=(K, R)$$

B是一种数据结构，它由数据元素的集合K和K上二元关系的集合R所组成。其中：

$$K=\{k_i \mid 0\leqslant i\leqslant n\}$$

$$R=\{r_j \mid 1\leqslant j\leqslant m\}$$

式中，k_i表示第i个数据元素；n为B中数据元素的个数，如果n=0，则K是一个空集，因此也就无结构而言，这是数据结构中的一种特殊情况；r_j表示第j个关系；m为K上关系的个数。

K上的一个二元关系r是序偶的集合。对于r中的任一序偶<x，y>(x，y∈K)，把x叫做序偶的第一元素，把y叫做序偶的第二元素，又称序偶的第一元素为第二元素的前驱，称第二元素为第一元素的后继。如在<x，y>的序偶中，x为y的前驱，而y为x的后继。

数据结构还能利用图形形象地表示出来，图形中的每个结点对应着一个数据元素，两结点之间带箭头的连线(称为弧或有向边)对应着二元关系中的一个序偶，其中序偶的第一元素为弧的起始结点(弧尾)，第二元素为它的终止结点(弧头)。

根据数据元素之间关系的不同特性，通常有下列4类基本结构：

(1)集合

结构中的数据元素之间除了“同属于一个集合”的关系外，没有其他的关系。

(2)线性结构

结构中的数据元素之间存在着一对一的关系。

(3)树形结构

结构中的数据元素之间存在一个对应多个的关系。

(4)图状结构或网状结构

结构中的数据元素之间存在多个对多个的关系。

上述4类基本结构的关系如图1.1所示。

从图状结构、树形结构和线性结构的定义可知，树形结构是图状结构的特殊情况，线性结构是树形结构的特殊情况。为了区别于线性结构，把树形结构和图状结构称为非线性结构。

下面用实际的例子来说明数据结构的概念。

表1.1是计算机系人事简表，表中共有10条记录，每条记录由6个数据项组成，由于每条记录的职工号各不相同，所以可以把每条记录的职工号作为该记录的关键字，在下面的例子中，将用记录的关键字来代表整个记录。

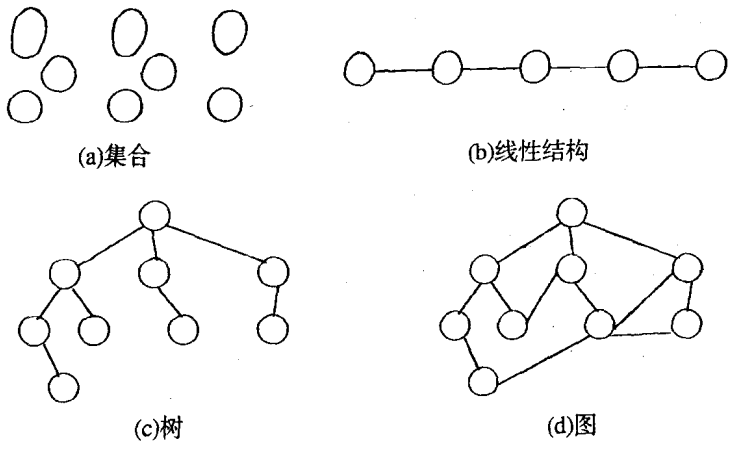


图 1.1　4 类基本结构关系图

表1.1　计算机系人事简表

职工号	姓　名	性别	出生年月	职　称	职　务	单　　位
01	王利华	男	1953年8月	教　授	系主任	
02	张 丽	女	1960年4月	讲　师	主　任	办公室
03	赵 军	男	1956年5月	副教授	主　任	计算机专业教研室
04	许渝川	男	1963年4月	讲　师	主　任	计算机基础教研室
05	孙　正	男	1950年9月	副教授		计算机专业教研室
06	张利敏	女	1966年5月	讲　师		计算机专业教研室
07	邓　折	男	1962年9月	讲　师		计算机基础教研室
08	王　涵	女	1957年1月	助　教		计算机专业教研室
09	郑　奇	男	1970年2月	助　教		计算机基础教研室
10	田　华	女	1975年4月			办公室

例1　对计算机系人事简表定义一种数据结构LIST=(K，R)，其中：

K={01，02，03，04，05，06，07，08，09，10}

R={<05，01>，<01，03>，<03，08>，<08，02>，<02，07>，<07，04>，<04，06>，<06，09>，<09，10>}

这一数据结构可以表示为：

05→01→03→08→02→07→04→06→09→10

可以发现，关系R是按职工年龄从大到小排列的关系。

在LIST中，除第一个元素05和最后一个元素10外，每一个数据元素有且仅有一个前驱元素，有且仅有一个后继元素，数据元素之间是一对一的关系，是线性结构。

例2 对计算机系人事简表定义另一种数据结构TREE=(K，R)，其中：

K={01，02，03，04，05，06，07，08，09，10}

R={<01，02>，<01，03>，<01，04>，<02，10>，<03，05>，<03，06>，<03，08>，<04，07>，<04，09>}

这一数据结构对应的图形如图1.2所示。

可以发现，关系R是人员之间领导与被领导的关系。

图1.2是一棵倒着画的树，在这棵树中，最上面的一个没有前驱、只有后继的结点称为根结点，最下面一层只有前驱、没有后继的结点叫做叶结点，除根结点和叶结点之外的结点叫做树枝结点。在一棵树中，除根结点外，每一个结点有且只有一个前驱结点，但可以有任意多个后继结点，这种数据结构的特点是数据元素之间一对多的联系，即1:N(N≥0)，这种数据结构是树形结构。

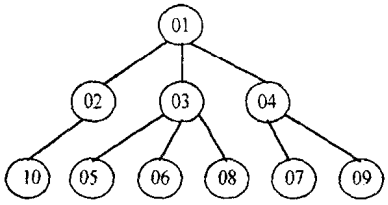


图 1.2

数据结构定义中的“关系”描述的是数据元素之间的逻辑关系，因此又称为数据的逻辑结构。为了在计算机中实现对数据结构的操作，必须研究数据结构在计算机中的表示。数据结构在计算机中的表示称为数据的物理结构(又称为映像)，亦称为存储结构，包括数据元素的表示和关系的表示。

数据元素之间的关系在计算机中有两种不同的表示法：顺序映像和非顺序映像，并由此得到两种不同的存储结构：顺序存储结构和链式存储结构。顺序映像的特点是借助于元素在存储器中的相对位置来表示数据元素之间的逻辑关系。非顺序映像的特点是借助于指示元素存储地址的指针表示数据元素之间的逻辑关系。在本书后面的内容中将结合具体的数据结构进行详细的讨论。

1.2 算法和算法评价

进行程序设计时，除了需要某种程序设计语言外，最主要的是要掌握算法的设计、分析和数据结构的有关知识，瑞士的N·沃斯教授提出了如下的公式：

算法＋数据结构=程序

并以此作为他所著的一本书的书名。这个公式说明了算法和数据结构对于程序设计的重要性，也说明了算法和数据结构二者之间的密切关系。

1.2.1 算法

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中的每一条指令表示计算机的一个或多个操作。算法具有下列5个重要特性：

(1)输入性　一个算法具有零个或多个输入。

(2)输出性　一个算法具有一个或多个输出。

(3)有穷性　一个算法必须在执行有穷步之后结束，而且每一步都可以在有穷时间内完成。

(4)确定性　算法中的每一条指令有确切的含义，无二义性。

(5)可行性　算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。

1.2.2 C语言的数据类型

本书在进行算法描述时，使用C语言作为工具。这里对C语言的数据类型作一个简要概述。

1.基本数据类型

C语言有3种由计算机硬件直接支持实现的基本数据类型：int型、float型和char型。

(1)char型存储空间为一个字符。

(2)float型有3种形式：float、double和long double。存储空间分别为4字节、8字节和16字节。

(3)int型可以有3个限定词：short、long和unsigned。short或long缩小或扩大了int型整数的存储空间和数值表示范围，由具体的C语言版本和机器确定。unsigned int表示其变量只能取正整数。

2.指针类型

C语言允许直接对存储变量的地址进行操作。存放地址的变量称为指针变量。例如定义int p，则&p表示变量p的地址，也可以称为指向变量p的指针。

在C语言中可以使用指针在函数之间传递数值。对于一个形参为简单变量的函数，参数的传递是值传送的形式，也就是说，把实参的值复制到形参，函数内形参的值被修改时，实参的值不会跟着变化。如果函数采用指针作为形参，那么参数传递采用地址传递的形式，即把实参的地址传给形参，这时如果函数内形参的值改变，则实参中的值跟着变化。可以有一维数组和多维数组。

3.数组类型

数组是同一类型的一组有序数据的集合，数组名标识了这一组数，数组元素的下标指示了数组元素在该数组中的顺序位置。数组下标的最小值称为下界，总是0。数组下标

的最大值称为上界，为数组定义值减1。如定义int a[100]，则下界为0，上界为99。

一个数组名代表了这一组数的起始地址，也就是说，数组名是一个指针变量。当数组作为函数参数时，其传递方法和指针作为函数参数时的传递方法相同，即采用传地址的方法。

C语言中系统并不检查数组下标是否越界，需要用户使用数组时自己注意不要越界。

4. 字符串

C语言中没有单独的字符串类型，把字符串定义成字符数组。每个字符串由转义符'\0'表示其结束。一个字符串常量由一对双引号表示。当一个字符串常量被存入内存时，系统自动在其末尾添加'\0'。

5. 结构体类型

结构体由一组称为结构体成员的项组成，每个结构体成员都有自己的标识符。定义一个结构体类型变量分为两步。第一步，定义结构体类型；第二步，定义结构体类型变量。例如：

```
struct  student                          /*    定义结构体类型       */
{
    long        no;
    char        name[40];
    char        sex[2];
    int         age;
    float       score;
};
struct  student  s1,s2[60],  *sp;        /*  定义结构体类型变量    */
```

结构体变量可以作为函数参数和函数返回值。

6. 共用体类型

共用体是把不同的成员项组织为一个整体，它们在内存中共用一段存储单元，但不同的成员项以不同的方式被解释。共用体类型的定义和操作方法与结构体类型变量相同。例如：

```
union  student              /* 定义共用体类型    */
{
    long        stu_no;
    char        stu_name[20];
};
union  student  stu;        /*  定义共用体类型变量    */
```

一个共用体变量不能同时存储多个成员项的值，它只能存储其中最后赋予它的值。

7. 枚举类型

枚举类型变量的取值只能是指定的若干个值之一。定义一个枚举类型变量由两步组

成：枚举类型定义和枚举类型变量定义。例如：

```
enum  color          /*   定义枚举类型          */
{
    red,green,blue,white,black,yellow;
};
enum  color   c1,  c2;          /*  定义枚举类型变量          */
```

枚举变量c1和c2只能取枚举类型定义中的6个值之一。枚举类型主要用于提高程序的可读性。

8.类型定义

C语言提供了产生新的数据类型名的功能，称为typedef。它并不引入一种新的类型，本身并不分配存储空间，它用另一个名称来描述已有的名称。例如：

```
typedef   int   INTEGER;
typedef   struct classmates
{
    int          no;
    char         name[40];
    char         address[200];
    struct classmates  *next;
} MATES;
```

这样就可以用INTEGER类型名来代替int类型名，用MATES来代替struct classmates类型名。

使用类型定义后，使依赖于机器的程序参数化，便于移植，可以为程序提供较好的说明信息，更便于理解，增加程序的可读性。

1.2.3 算法评价

一个“好”的算法应满足以下目标：

(1)正确性

正确性是设计和评价一个算法的首要条件，要求该算法在合理的输入数据下，能在有限的时间内得出正确的结果。

(2)可读性

算法不但要转变成机器可执行的程序，也为了人的阅读与交流。可读性好有助于人对算法的理解，也有助于算法中隐藏错误的排除。

(3)健壮性

当输入数据非法时，算法应能作出适当的处理，不会产生不可预料的结果。

(4)高效率

算法的效率是指算法的执行时间。如果对于同一个问题有多个算法可供选择，应尽