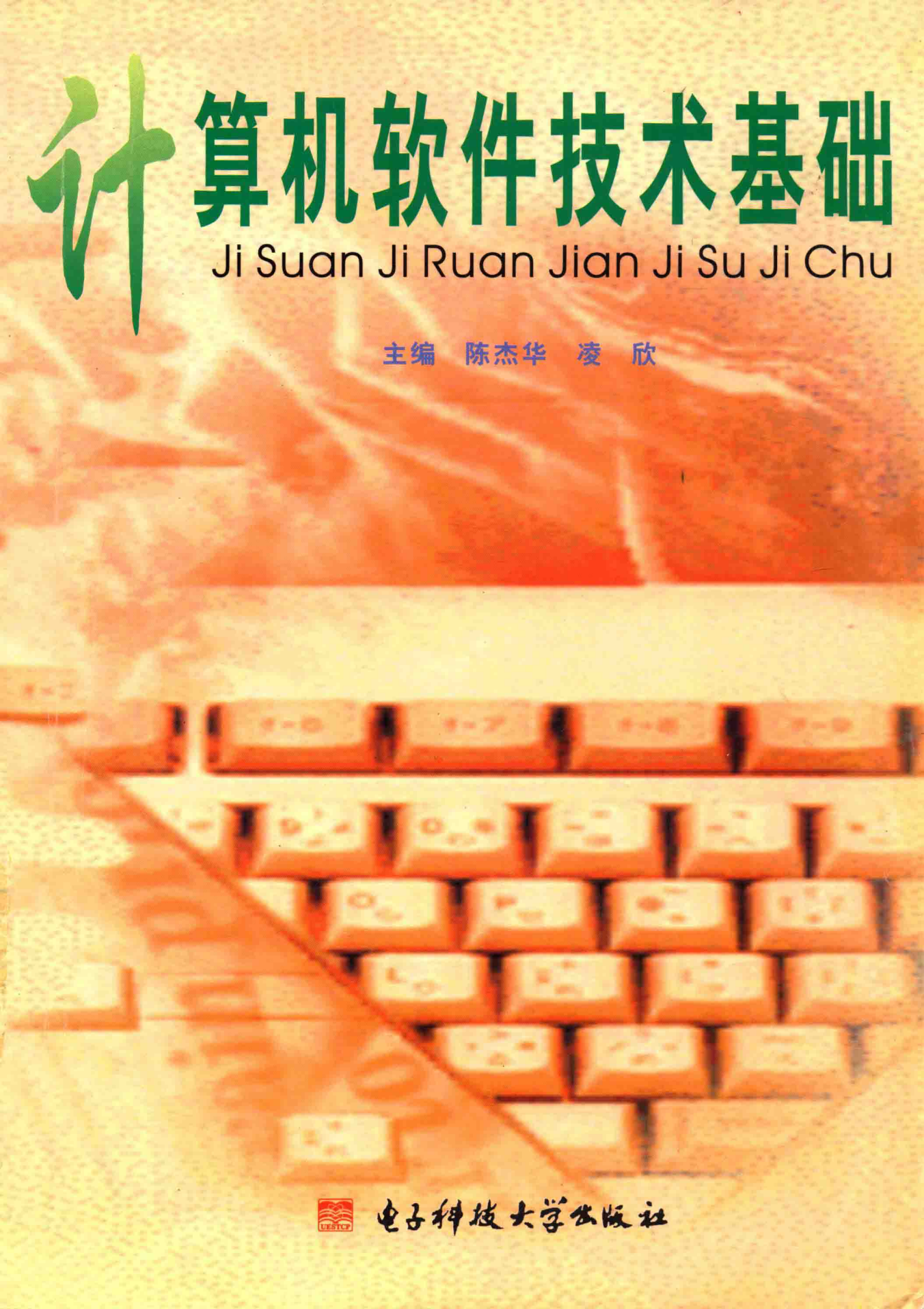
计算机软件技术基础

Ji Suan Ji Ruan Jian Ji Su Ji Chu

主编 陈杰华 凌 欣



电子科技大学出版社

计算机软件技术基础

Ji Suan Ji Ruan Jian Ji Shu Ji Chu

计算机软件技术基础

主编　陈杰华　凌　欣

电子科技大学出版社

图书在版编目（CIP）数据

计算机软件技术基础/陈杰华主编 .—成都：电子科技大学出版社，2001.8

ISBN 7-81065-709-7

Ⅰ.计算… Ⅱ.陈… Ⅲ.软件—基础知识 Ⅳ.TP31

中国版本图书馆 CIP 数据核字（2001）第 050296 号

内 容 提 要

随着计算机软硬件技术的迅速发展与普遍推广，社会各界对从事计算机应用的工程技术人员提出了更高的要求，计算机软件技术已成为各类高等工科院校非计算机专业学生必须具备的一项技术基础。本书主要介绍有关计算机软件技术的基础知识，如基本数据结构、操作系统基础、软件工程技术、数据系统原理与应用等，注重介绍基本知识、基本理论、使用方法和编程手段。在每章的最后，附有丰富的习题以供读者学习时参考。

本书内容安排合理，文字简洁易懂，紧扣课程要求。对参加计算机等级考试三级（B）的考生是必备的考前应试辅导材料，本书也可供大、中专院校计算机专业学生作为学习《软件技术基础》课程的参考书籍。

计算机软件技术基础

主编 陈杰华 凌 欣

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：谢应成
发　　行：新华书店经销
印　　刷：四川导向印务有限公司
开　　本：787×1092　1/16　印张 18　字数 438 千字
版　　次：2001 年 8 月第一版
印　　次：2001 年 8 月第一次印刷
书　　号：ISBN 7-81065-709-7/TP·475
印　　数：1—4000 册
定　　价：25.00 元

前　言

本书根据教育部关于全国高等工科院校非计算机专业《软件技术基础》课程的基本要求，并参考许多省、市非计算机专业计算机等级考试三级大纲（B）的要求，结合本书作者多年的教学实践编写而成的。

全书内容包括：

1. 基本数据结构

如线性表、栈、队列的存储结构及算法、查找、排序等。

2. 操作系统基础

如操作系统概念、处理机管理、作业管理、存储管理、文件管理、设备管理、网络操作系统、Windows NT 操作系统简介、Unix 操作系统简介等。

3. 软件工程思想和方法

如软件的概念和特点、软件的发展过程和软件危机、软件工程及软件工程学、软件工程模式、软件生存周期、软件需求分析、软件设计等。

4. 数据库系统原理

如数据库系统的概念、数据模型、数据库系统的体系结构与数据独立性、实体一联系模型、关系模型、关系数据库管理系统、关系数据库标准语言 SQL、数据库设计等。

本书的编写得到四川大学制造学院的领导和同仁的关心与支持，尤其是工程设计中心的教师和实验人员提出了许多宝贵建议，在此一并表示感谢。

全书共分为四章，第一章和第二章由陈杰华编写，第三章和第四章由凌欣编写，第三章和第四章的全部插图由陈林绘制。另外，参与本书教学讨论、文字录入、资料整理、绘制图片、编辑校对等工作的有方铸成、钟华庆、陈若茜、蒋皖、罗雪、陈烈、刁倩玲、李志存、刘济民、匡亚都、陈海英、刘思源、刘西蜀、张光辉、陈若林，在此向他们表示衷心地感谢。

由于编者水平有限，加之编写出版时间极为紧迫，书中难免存在许多不妥之处，恳请读者批评和指正。

编　者

2001 年 6 月

目　录

第一章　基本数据结构

《数据结构》是随着计算机科学技术发展而形成的一门新学科，它是计算机科学与技术专业本科生的一门重要基础课程，也是许多非计算机专业本科生的一门必修课程。这门课程的基本内容不仅是一般程序设计方法学（尤其是非数值运算程序设计）的重要基础，而且还是学习编译程序、操作系统、数据库系统、文字处理系统等课程的必备基础，也是设计和实现其它系统程序和应用程序的重要基础。

《数据结构》课程要讨论的主要问题就是根据实际应用问题的要求，合理地组织并存储各种非数值运算数据，从而设计出相应的计算机算法。

1.1　绪　　论

1.1.1　计算机应用与非数值运算

1.　计算机应用

计算机软件技术与硬件技术的迅猛发展，不仅开创了科学技术发展的新纪元，同时也给人类社会的科学技术进步带来巨大的影响和推动。从日常生活到工业生产，从文化教育到尖端科技，我们都可以使用计算机来进行操作。

计算机的应用领域主要可以归纳为如下四个方面。

（1）科学计算

在科学研究和工程设计中，需要进行大量复杂的高精度数值计算、数理统计、结构计算、模拟分析等。

（2）数据处理

指对大量的数据进行加工处理，如情报、档案、图书等信息的检索，排版印刷、办公自动化、生产、物质、财务、人事等管理都可采用计算机来进行处理。

（3）实时控制

指计算机及时采集数据并进行处理，按最佳方式迅速地对控制对象加以控制。例如航天飞行、宇航空间站发射、对接、测控等。利用实时控制可以代替人进行各种有害的、危险的现场操作与控制。

（4）辅助设计

利用计算机辅助设计可以帮助人们进行汽车、船舶、建筑、化工、大规模集成电路以及计算机本身的设计自动化，还可以进一步促进人工智能技术的发展和应用。计算机辅助设计的主要内容如表 1.1 所示。

表 1.1　计算机辅助设计

英文缩写	英 文 全 称	说　明
CAD	Computer Aided Design	计算机辅助设计
CAT	Computer Aided Test	计算机辅助测试
CAM	Computer Aided Management	计算机辅助管理
CAI	Computer Aided Instruction	计算机辅助教育

2. 非数值运算

从计算机的发展过程来看，电子计算机问世的直接原因只是为了解决弹道学方面的许多数值计算问题。电子计算机的早期应用范围，主要也就局限在科学计算和工程计算两个方面，当时计算机的处理对象完全是纯粹的数值运算数据，所以人们通常将这类科学运算问题称为数值计算。

50 多年以来，计算机科学和技术的发展异常迅猛。一方面表现在计算机的运算速度不断提高，信息的存储容量日益增大，数据精度越来越高，价格越来越便宜；另一方面计算机应用范围已经远远超出科学计算与工程计算的领域，可以说计算机已经渗透到人类社会活动的一切领域，从日常生活到工业生产，从文化教育到尖端科技，我们都可以看到许多计算机应用实例。这时，计算机的处理对象也就从传统的纯粹数值运算数据发展到有一定结构特征的非数值运算数据。

学习过《程序设计语言》课程的同学都知道，数据的组织结构和表示方式都将直接关系到计算机算法的数据处理效率。另外，在现在的计算机应用中，许多系统程序和应用程序的规模越来越大，数据结构越来越复杂，处理对象又都是非数值运算数据。所以，一个程序员只依靠自身的编程经验和解题技巧是不可能设计出优秀应用程序的。要更好地进行程序设计，并有效地使用并发挥计算机系统的数据处理效率，程序员就需要对应用程序加工处理的对象进行系统研究，理解非数值运算数据的特性以及各种非数值运算数据之间存在的相互关系。

1.1.2　数据、数据项、数据元素与数据结构

1. 数据

所谓数据（data）就是计算机程序所处理的一切数值性的和非数值性的信息，以及各种系统程序和应用程序。由于计算机是一种现代化的信息处理机器，所以在计算机科学中的数据含义是非常广泛的。例如，数值、字符等信息是数据，声音、图像等信息也是数据，甚至程序本身都是数据。这是因为它们都可以按照一定的数据格式输入计算机，并由计算机程序进行加工处理。

2. 数据项

所谓数据项（data item）就是数据不可再分的最小单位。

注意：*在有些情况下，数据项又可以被称为域、字段等。*

3. 数据元素

所谓数据元素（data element）就是组成数据的基本单位。一般情况下，一个数据元素

是由一个或多个数据项组成的。例如：

- 对数组而言，每一个数组元素就是数组中的一个数据元素。
- 对数据文件而言，每一个记录（record）就是数据文件的数据元素。其中，每个记录由若干个固定的数据项（域或字段）组成。

注意：在有些情况下，数据元素又可以被称为结点、节点、记录等。

4. 数据结构

在介绍数据结构（data structure）概念之前，我们首先分析一个学生花名册，以便说明数据结构的准确含义。

假定有一个学生花名册，记录某个大学全体学生的姓名和相应的住址。现在要求写一个查找算法，具体要求如下：在指定任何一个学生姓名后，该查找算法能够查找出相应学生的住址。很明显，查找算法的设计将完全依赖于花名册中学生姓名及相应的住址是如何组织、计算机又是如何存储花名册信息的。

- 顺序数据组织：花名册中的学生姓名和相应住址内容是任意排列的，即排列次序没有任何规律可言。这样当指定一个学生姓名后，查找算法只能对花名册从头开始与给定的学生姓名逐个进行比较，直至查找到所给定的学生姓名时为止。这种查找方法效率非常低，查找过程也是相当费时的。
- 索引数据组织：花名册中的学生姓名和相应住址内容是进行适当组织过的。一方面，学生姓名和相应住址内容是按学生所在班级来排列的，另一方面再建立一个索引表用于登记每个班级的学生信息在花名册中的开始位置。这样当要查找某个学生的住址内容时，则可以首先从索引表中查到该学生所在班级的学生信息是从什么位置开始的，然后就从该开始位置进行查找，而完全没有必要去查找其它班级的学生信息。

下面我们对学生花名册的使用情况作进一步分析。

- 当有一个新学生进入大学时，花名册需要增加新学生的姓名和相应住址内容。对于上面的两种具体数据组织形式，如何实现插入（insert）操作？是将要增加的学生姓名和住址内容插在花名册的前头，还是插在花名册的末尾，或是插在花名册中间的某个位置上？插入操作完成后，对花名册的原有数据组织形式是否有影响？
- 当有一个老学生离开大学时，应从花名册中删除该离开学生的姓名和相应的住址内容。删除（delete）某个学生的姓名和住址内容后，其它学生的姓名和住址内容是否要移动？如果需要移动则应该如何移动？

上面两个问题说明，为适应数据增加和减少的需要，计算机算法还必须对数据结构定义一些相应的运算。上面只涉及到两种运算，即插入运算和删除运算。当然，还可以提出一些其它可能的运算来满足实际应用的需要，如某个学生的住址发生变化后，为处理这种变化，就应该定义一个修改（modify）运算。

实际上，学生花名册就是一个数据结构问题。我们从中可以看到，计算机算法与数据结构两者之间是紧密相关的，即计算机算法完全依附于具体的数据结构，数据结构将直接关系到算法选择和实现效率。改变数据的组织形式，或者说使用新的数据结构，可以编写出一个效率完全不同的计算机算法。对数据结构定义的任何运算都是由计算机来实现的，

所以需要为数据结构设计相应的计算机算法。换言之，数据结构设计还需要给出每种数据结构所定义的全部运算算法。

所谓数据结构（data structure）就是研究数据元素之间抽象化的相互关系和这种相互关系在计算机系统中的存储方式，并对每种数据结构定义相关的运算算法，确保经过运算后仍然是原来的数据结构类型。

- 数据的逻辑结构（logical structure）：该结构用于表示数据元素之间抽象化的相互关系，又可简称为数据结构。
- 数据的物理结构（physical structure）：该结构用于表示数据元素之间抽象化的相互关系在计算机系统中的存储方式，又可简称为存储结构（storage structure）。

1.1.3　算法的五大特性

算法（algorithm）一词是非常古老的，一次宗教仪式、一张医院处方、一个学习计划等都可以认为是一个算法。在计算机科学中，所谓算法就是为解决某个计算任务而安排的有限操作过程的描述。通常，一个计算机算法必须具有如下五大特性。

1. 有穷性

一个算法最基本的特性就是有始有终，不管具体情况如何，一个算法都必须在执行有穷次操作后结束。不过，一些算法操作过程的有限并不能保证计算机执行环境的有穷性。所以，判断一个算法的有穷性应该对计算机执行情况进行分析，从而才能作出正确的判断。下面，我们考查两个计算机算法。

下面是一个错误的算法。

第 1 步　　算法开始
第 2 步　　计算 n = 0
第 3 步　　计算 n = n + 1
第 4 步　　重复执行第 3 步
第 5 步　　算法结束

说明：

- 从表面上看，该算法只有五个步骤。实际上，该算法将永无休止地执行下去，直到计算机发生溢出（overflow）时为止。
- 由于该算法不具备有穷性这一特性，所以它不是一个正确的算法。

下面是一个正确的算法。

第 1 步　　算法开始
第 2 步　　计算 n = 0
第 3 步　　计算 n = n + 1
第 4 步　　如果 n 的值为 100，则执行第 5 步，否则重复执行第 3 步。
第 5 步　　输出 n 的值
第 6 步　　算法结束

2. 确定性

算法中的每一个操作都应该是确定的，而不能出现任何歧义性。

3. 能行性

算法中的每一个操作都应该是有效的，并得到确定的结果。

4. 数据输出

一个算法必须有一个或多个数据输出，用于表示计算机执行的结果。所以，没有数据输出的算法是完全无用的。

5. 数据输入

一个算法有 n（n≥0）个初始数据的输入，从而使算法能够通用。

1.1.4 算法描述语言和算法分析

1. 算法描述语言

研究数据结构的主要目的就是为了编写出优秀的计算机算法，尤其是非数值运算方面的计算机算法。所以，在后面讨论每种数据结构时，都将给出相应的基本运算算法。在描述一个算法时，我们可以使用自然语言、传统流程图、结构化流程图（N－S 图）、伪代码等。大多数情况除使用文字进行必要的叙述外，还将使用一种程序设计语言来描述算法。

要使算法描述明确、清晰和简洁，便于程序员理解与掌握算法的思想和实质，必须十分慎重地选择描述算法的程序设计语言工具。基于 ANSI C 是当今最流行的计算机程序设计语言，它具有丰富的数据类型，能较好地体现结构化（structured）程序设计思想。所以，本章将完全采用 ANSI C 语言来描述数据结构及其相关的算法。

下面，对算法的描述语言 ANSI C 作如下说明。

（1）用函数形式表示算法

所有算法的具体函数形式如下：

```
类型标识符 函数名（形式参数表）
{
    说明部分
    语句部分
}
```

说明：

- 类型标识符：用于指定函数返回值的数据类型，如果函数没有返回值，则类型标识符应该使用 ANSI C 语言中的关键字 void。
- 语句部分：这是由一个或多个 ANSI C 语句构成的，用于说明函数中所用的参数。
- 形式参数表：可以包含有若干个形式参数，如果形式参数多于一个，则彼此间要使用逗号分隔。形式参数的数据类型说明采用 ANSI C 标准，即在列出形式参数的同时应该说明相应的数据类型，例如 int max(int x，int y)。

（2）数据元素的类型

数据元素的类型约定为 datatype，这是由用户在使用该数据类型时自行定义的。

（3）常量定义

在计算机算法中，使用 NULL 表示空指针，使用标识符 MAXSIZE 表示指定的数组元素个数。如果没有预先定义 MAXSIZE 的具体内容，则约定：

```
#define MAXSIZE 100
```

（4）数据结构的表示

数据结构的表示可用结构体类型进行说明如下：

```
typedef struct 结构体标识符
{
    类型名 1  结构成员名表 1;
    类型名 2  结构成员名表 2;
        …
        …
    类型名 N  结构成员名表 N;
}   类型标识符;
```

说明：

- 关键字 typedef：用于定义一种新的类型名称，原有的类型名称依然有效。
- 关键字 struct：用于表示结构体类型，这是用户自定义的标识符。

（5）函数调用

调用函数的一般形式如下：

函数名（实在参数表）

说明：

- 实在参数表中的实际参数应与被调用函数形式参数表中的形式参数相对应，并且数据类型必须是一致的。

（6）参数传递

在进行函数调用时，实在参数的值被传递给形式参数。由于 ANSI C 语言规定，当函数的参数为非数组型变量时，参数传递为单向的“值传递”方式，即只能将实在参数的值传递给形式参数，而不能将形式参数的值传递给实在参数。所以，形式参数值的改变并不会影响相应实在参数的值。要使被调用函数能直接改变非数组型实在参数的值，则应该采用双向的“地址传递”方式，即将变量的地址作为实在参数传递给形式参数，而被调用函数中相应的形式参数应该说明为指针类型。这样，形式参数和实在参数将占用同一个内存地址，因此形式参数值的改变才会影响相应实在参数的值。

2. 算法的复杂性分析

在计算机程序设计中，算法的的复杂性分析是十分重要的，这也是每个程序员都应该掌握的基本技术。对于一个给定计算任务的求解，程序员往往可以提出若干个计算机算法，那么如何评价一个算法的优劣呢？如何从多个算法中找出最佳的算法呢？如何将一个新算法与其它算法进行比较呢？这些就是算法复杂性分析要讨论的问题。

评价计算机算法优劣的标准很多，通常是以执行算法所需要的机器时间和所占用的存储空间来判断一个算法的优劣。实际上，算法的执行时间和所需要的存储空间往往是相互矛盾的，两者往往难以兼得。所以，对于具体计算任务就应该进行具体分析并合理取舍。由于硬件价格越来越低，使得存储空间的花费不断下降。因此，大多数情况下可以使用算法的执行时间作为主要的评价标准。

一个算法的执行时间大致等于其所有语句执行时间的总和，而任何一条语句的执行时间为该语句的执行次数与执行一次该语句所需时间的乘积。由于每一条语句执行一次所需的时间都是由机器的软硬件环境决定的，与计算机算法无关。所以，算法分析只能粗略地

对算法中语句的执行次数作出估计，以便得到算法的执行时间，而不能精确地计算出算法的具体执行时间。

使用 ANSI C 语言描述两个 n 阶矩阵相乘的算法。

```
1   for (i=0;i<n;i++)
2   for (j=0;j<n;j++)
3   {
4       c[i][j]=0;
5       for (k=0;k<n;k++)
6           c[i][j]=c[i][j]+a[i][k]*b[k][j];
7   }
```

该算法每一行的具体执行次数如表 1.2 所示。

表 1.2　算法的具体执行次数

行　数	执 行 次 数
1	n 次循环控制
2	n^2 次循环控制
4	n^2 次赋值运算
5	n^3 次循环控制
6	n^3 次赋值运算，其中包括赋值运算、加法运算和乘法运算

尽管 for 循环控制语句是由若干条语句组合而成的，但我们在这里将它当作一条简单语句看待，这样可以使计算任务得到简化，同时又不影响算法分析的最终结果。上述算法中的总语句执行次数 x_n 为：

$$x_n=2n^3+2n^2+n \tag{1.1}$$

很明显，总语句执行次数 x_n 是关于 n 的一个非线性函数。n 是给定计算任务的规模（如矩阵阶、线性表长度、树的顶点个数、数组元素个数等）。对算法分析而言，最重要的是确定 x_n 是 n 的什么函数，进而分析 x_n 随数据 n 的变化情况，并最终确定 x_n 的数量级。

说明：

- 语句“x=x+1”的执行时间数量级为 O(1)。
- 语句“for (k=0; k<=n; k++) x=x+1”的执行时间数量级为 O(n)。
- 语句“for (j=0; j<=n; j++) { for (k=0; k<=n; k++) x=x+1}”的执行时间数量级为 $O(n^2)$。

注意：随数据 n 的增大，总语句执行次数 x_n 增长速度较慢的算法就是最优算法。

如果我们用希腊字母“O”来表示数量级，则（1.1）式可以写成：

$$x_n=O(n^3) \tag{1.2}$$

（1.2）式表示执行次数 xn 将不超过 n^3 的数量级，即 x_n 的数量级与 n^3 成正比。在算法分析过程中，对一个算法可以将每一条语句的执行次数进行分析，并求出其总的执行次数，从而得到一个多项式：

$$xn=C_kn^k+C_{k-1}n^{k-1}+\cdots+C_1n+C_0 \tag{1.3}$$

这里的 C_i（$0\leqslant i\leqslant k$）为常数，且 $C_k\neq 0$，（1.3）式中的每一项反映不同语句执行次数的数量级。我们取其中的最高数量级作为整个算法的数量级，换言之，将算法中那些具有最大执行次数语句的数量级作为整个算法的数量级。所以：

$$x_n=O(n^k) \tag{1.4}$$

以此就可以判断一个计算机算法的执行时间，并称算法的执行时间为 $O(n^k)$。在本章内容中，所使用的算法执行时间数量级如表 1.3 所示。

表 1.3　算法执行时间数量级

数量级	说　明
O (1)	表示算法的执行时间为一个常数，这类算法为常数型
O (n)	表示算法的执行时间为一个变数，这类算法为线性型
$O(n^2)$	表示算法的执行时间为一个变数的二次运算，这类算法为平方型
$O(n^3)$	表示算法的执行时间为一个变数的三次运算，这类算法为立方型
$O(2^n)$	表示算法的执行时间为一个变数的指数运算，这类算法为指数型
O (log n)	表示算法的执行时间为一个变数的对数运算，这类算法为对数型

注意：如果一个算法的执行时间为 O (log n)，则当 n 充分大时它比 O (n)的速度要快得多，但比 $O(n^2)$的速度要慢得多。

1.2　线 性 表

线性表（linear list）是一种非常简单的常用数据结构，它的存储结构主要有两种，即顺序存储结构和链式存储结构。本节将首先讨论线性表的定义、线性表的两种存储结构和线性表的基本运算及其实现算法，最后讨论线性表的应用实例。

1.2.1　线性表概念与运算

线性表具有以下两大特性：

- 线性表中的全部数据元素，其数据类型是完全一致的。
- 每个数据元素在线性表中的位置只取决于它们本身的顺序号，数据元素之间的相对位置是线性的。所以，线性表是一种线性结构。

1. 线性表概念

所谓线性表就是由一组性质相同的数据元素组成的有限序列，其中的数据元素在不同情况下可以具有完全不同的含义。

- 阿拉伯数字（0，1，2，3，4，5，6，7，8，9）是一个线性表，表中的数据元素是单个数字，共有 10 个数据元素。
- 每周的全部工作日（星期一，星期二，星期三，星期四，星期五）是一个线性表，表中的数据元素是一个工作日，共有 5 个数据元素。
- 我国八大地区网络中心的城市名称表（北京，沈阳，上海，西安，南京，成都，