21世纪高等学校计算机科学与技术专业规划教材

数据结构

徐翠霞　崔玲玲　主　编
孙学农　张冰川　陈　姗　副主编

中国电力出版社
www.infopower.com.cn

21世纪高等学校计算机科学与技术专业规划教材

数据结构

徐翠霞 崔玲玲 主 编
孙学农 张冰川 陈 姗 副主编
田 华 祝惠新 参 编

中国电力出版社
www.infopower.com.cn

内容提要

本书系统地介绍了各种常用的数据结构，内容丰富，概念讲解清楚，叙述严谨流畅，逻辑性强。书中配有大量图表、丰富的例题和习题，对给出的每一种算法，均先描述了它的基本思路和要点，使得算法清晰易读，便于学生理解和掌握。本书有配套出版的《数据结构上机实验与习题解析》，既便于教学，又便于自学。

本书共分 9 章，主要包括绪论、线性表、栈和队列、串、多维数组和广义表、树和二叉树、图、查找、排序等内容。

本书可作为本科和高职高专院校计算机类专业或信息类专业的教材，也可供广大从事计算机工程与应用的科技工作者参考。

图书在版编目（CIP）数据

数据结构/徐翠霞，崔玲玲主编．—北京：中国电力出版社，2006

21 世纪高等学校计算机科学与技术专业规划教材

ISBN 7-5083-4282-8

I. 数... II. ①徐...②崔... III. 数据结构-高等学校-教材 IV. TP311.12

中国版本图书馆 CIP 数据核字（2006）第 047159 号

丛 书 名：21 世纪高等学校计算机科学与技术专业规划教材

书　　名：数据结构

出版发行：中国电力出版社

地　　址：北京市三里河路 6 号　　邮政编码：100044

电　　话：（010）68362602　　传　　真：（010）68316497，88383619

本书如有印装质量问题，我社负责退换

服务电话：（010）88515918（总机）　　传　　真：（010）88518169

E-mail：infopower@cepp.com.cn

印　　刷：汇鑫印务有限公司

开本尺寸：185×260　　**印　　张**：15.25　　**字　　数**：372 千字

书　　号：ISBN 7-5083-4282-8

版　　次：2006 年 6 月北京第 1 版

印　　次：2006 年 6 月第 1 次印刷

印　　数：0001—4000 册

定　　价：23.00 元

前　言

在计算机及其应用的各个领域中，都会用到各种各样的数据结构，学会分析研究计算机加工对象的特性，选择合适的数据结构和存储表示，以及编制相应的实现算法，是计算机工作者不可缺少的技能。因此，《数据结构》是高等院校计算机专业教学中的一门重要专业基础课。在我国当前的计算机专业教学计划中，它是核心课程之一。

本书介绍各种最常用的数据结构，阐述各种数据结构内涵的逻辑关系，讨论它们在计算机中的存储表示，以及在这些数据结构上的运算（操作）和实际的执行算法，并对算法的效率进行简要的分析和讨论。

本书内容根据教育部高等学校规划教材指导思想与原则的要求，充分考虑了学生的培养目标和教学特点。在内容的组织上，本着由浅入深、循序渐进的原则，注重基本知识和基本概念的介绍，结合实例重点介绍实用性较强的内容，对难度过大的内容只作少量介绍，使学生有的放矢，掌握所学内容。

本书共分 9 章：第 1 章绪论，主要介绍数据结构的基本概念和算法的基本知识；第 2 章至第 7 章主要介绍几种基本的数据结构，即线性表、栈、队列、串、多维数组、广义表、树、二叉树和图；第 8、9 两章分别介绍数据处理中广泛使用的技术——查找和排序。

本书既注重原理又重视实践，配有大量的例题和习题，解释比较详细。书中算法采用 C 语言描述，且均在计算机上运行通过，又做了较详细的注解，有利于读者理解算法的实质和基本思想。书中每一章均有小结，概述了该章的内容提要和学习要求，以便于读者学习和抓住重点。每章配置的习题可以检验读者的学习效果和培养程序设计的能力。

本书由徐翠霞、崔玲玲担任主编，孙学农、张冰川、陈姗担任副主编。第 6 章和第 8 章由徐翠霞编写；第 2、3、4 章由崔玲玲编写；第 7 章由孙学农编写；第 1 章和第 5 章由张冰川编写；第 9 章由陈姗编写。全书由徐翠霞负责统稿。另外，参与本书编写的还有田华和祝惠新，在此一并表示感谢。

由于作者水平有限，书中错误和不足之处在所难免，恳请广大读者批评指正。

作　者

2006 年 5 月

目　录

第1章 绪 论

本章学习目标

本章主要介绍数据结构课程中一些常用术语的定义，集合、线性结构、树和图等常用数据结构的表示，算法的时间复杂度和空间复杂度评价方法等内容。通过本章的学习，要求同学们：

- 熟悉各名词、术语的含义，掌握基本概念，特别是数据的逻辑结构和存储结构之间的关系。分清哪些是逻辑结构的性质，哪些是存储结构的性质；掌握集合、线性结构、树和图等常用数据结构的二元组表示；
- 掌握算法的特性和算法描述的方法；
- 掌握评价算法的一般规则，算法的时间复杂度的定义和数量级表示，时间复杂度在最好、平均、最坏三种情况下的含义，算法的空间复杂度的定义和数量级表示；
- 掌握计算语句频度和估算算法时间复杂度的方法。

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，它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或数据处理、过程控制以及对文件的存储和检索及数据库技术等，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。

1.1 基本概念和术语

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，仅掌握几种计算机程序设计语言是难以应付众多复杂的课题的。要想有效地使用计算机、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。打好“数据结构”这门课程的扎实基础，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。

在系统地学习数据结构知识之前，先了解一些基本概念和术语。

1. 数据

数据（Data）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，它是计算机程序加工的“原料”。常见的数据可分为数值数据和非数值数据。数值数据包括整数、实数，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、图像和语音等。

2. 数据元素

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。在不同的条件下，数据元素又可称为元素、结点、顶点、记录等。有时，一个数据元素可由若干个数据项（Data Item）组成，其中能够唯一标识一个数据元素的数据项称为关键字。数据项是数据不可分割的最小单位。

例如，学籍管理系统中学生信息表的每一个学生记录就是一个数据元素。它包括学生的学号、姓名、性别、籍贯、出生年月和成绩等数据项。通常，在解决实际问题时把每个学生记录当作一个基本单位进行访问和处理。

3. 数据结构

数据结构（Data Structure）是指按照某种关系组织起来的一组数据，按照一定的存储表示方式把它们存储在计算机中，并在这些数据上定义了一个运算的集合。数据结构一般包括以下三个方面的内容：数据的逻辑结构、数据的存储结构以及数据的运算。

（1）数据的逻辑结构。

对数据之间关系的描述称为数据的逻辑结构。在任何问题中，数据元素之间都不会是孤立的，在它们之间都存在着这样或那样的关系，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：在集合中，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。集合是元素关系极为松散的一种结构。②线性结构：该结构中的数据元素之间存在着一对一的关系。③树型结构：该结构的数据元素之间存在着一对多的关系。④图形结构或网状结构：该结构的数据元素之间存在着多对多的关系。图 1-1 为表示上述 4 类基本结构的示意图。

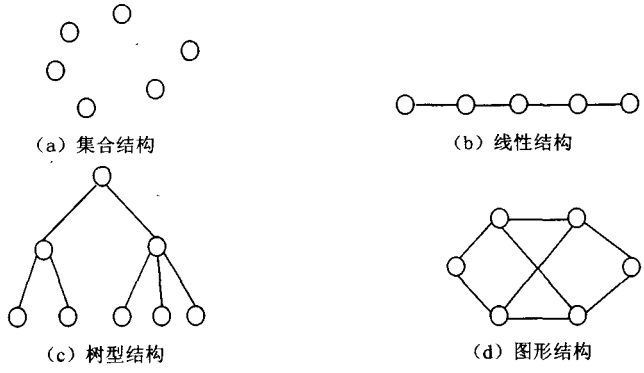

图 1-1　4 类基本结构的示意图

在形式上，数据的逻辑结构可以用一个二元组

$$Data\_Structure = (D, R)$$

来表示。其中，D 是数据元素的有限集，R 是 D 上关系的有限集。R 中每个关系都是从 D 到 D 的关系。

设 B= (K, R)是一个逻辑结构，r 是一个从 K 到 K 的关系，$r \in R$。若 $k, k' \in K$，且 $<k, k'> \in r$，则称 k'是 k 的后继，k 是 k'的前驱，这时 k 和 k'是相邻的结点或元素（相对关系 r 而言）。如果不存在一个 k'使 $<k, k'> \in r$，则称 k 为关于 r 的终端结点（即无后继的结点）。如果不存在一个 k'使 $<k', k> \in r$，则称 k 为关于 r 的开始结点（即无前驱的结点）。如果 k 既不是终端结点，

也不是开始结点，则称 k 是内部结点。

在本书所讨论的数据结构中，一般只讨论包含一个关系 r 的关系集 R，即 R={r}。对于 R 中包含多个关系的情况，可用类似的方法进行讨论。

例如，在计算机科学中，复数可取如下定义：

$$Complex = (C, R)$$

其中，C 是含有两个实数的集合$\{c_1, c_2\}$，R={r}，而 r 是一个从 C 到 C 的关系。在关系 r 中，只有一个有序对$<c_1, c_2>$，该有序对$<c_1,c_2>$表示 c_1 是复数的实部，c_2 是复数的虚部。

（2）数据的存储结构。

数据的逻辑结构是从逻辑关系上来描述数据，它与数据的存储无关，是独立于计算机的。而我们研究数据结构的目的是为了在计算机中实现对它的运算，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称为数据的物理结构，或称存储结构。它所研究的是数据结构在计算机中的实现方法，包括数据结构中元素的表示及元素间关系的表示。

数据的存储结构可以采用顺序存储方法和链式存储方法。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假设的指针来表示，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，有时为了查找的方便还采用索引存储方法和散列存储方法。索引存储方法是在存储元素信息的同时，尚需建立一个附加的索引表。散列存储方法是根据数据元素的值来确定该元素的存储位置。这两部分的有关内容将在第 8 章中详细讨论。

一般数据结构的存储表示都是这 4 种存储表示之一，或是它们的组合。同一个逻辑结构也可以有几种不同的存储表示方法，选择哪一种存储表示方法要根据运算的实际情况来具体确定。

数据的存储结构是逻辑结构在计算机中的实现，它是依赖于计算机的。

（3）数据的运算。

数据的运算是在数据上所施加的一系列操作。数据的运算是定义在逻辑结构上的，而运算的具体实现要在存储结构上进行。数据的各种逻辑结构有相应的各种运算，每个逻辑结构都有一个运算的集合。例如，最常用的运算有：检索（查找）、插入、删除、更新和排序等。

1.2 数据结构的重要性

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。当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个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。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。

由于当时所涉及的运算对象是简单的整型、实型或布尔类型数据，所以程序设计者的主要精力集中于程序设计的技巧上，而无须重视数据结构。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。这类问题涉及到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，而是如何设计出合适的数据结构。请看下面的 3 个例子。

【例 1-1】学生信息查询系统。

当我们需要查询某个专业或某个年级的学生信息时，只要建立相关的学生信息数据结构，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，可以在学生信息查询系统中建立一张按学号顺序排列的学生信息表和分别按姓名、专业、年级顺序排列的索引表，如图 1-2 所示。由这 4 张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。

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	000803
何文颖	000801
李淑芳	980002
刘　丽	990301 010601
石宝国	990303
魏永鸣	010602
吴承志	980001
赵胜利	000802
张会友	990302

（b）姓名索引表

计算机科学与技术	980001 990303 000801 010601
信息与计算科学	980002 990302 000803
数学与应用数学	990301 000802 010602

（c）专业索引表

2000 级	000801 000802 000803
2001 级	010601 010602
98 级	980001 980002 980003
99 级	990301 990302

（d）年级索引表

图 1-2　学生信息查询系统的数据结构

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，这类数学模型可称为线性结构。

【例 1-2】有线电视网收费转播系统。

某收费有线电视网计划转播一场重要的足球比赛。它们的转播网和用户终端构成一个树型结构，如图 1-3 所示。

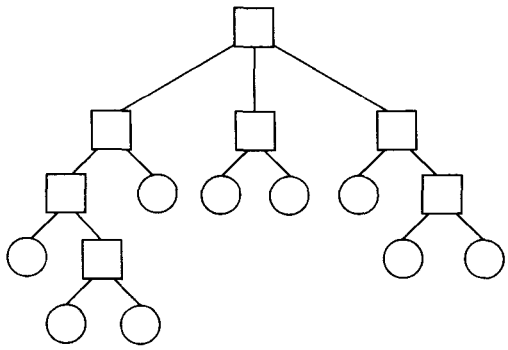

图 1-3　有线电视网收费转播系统的数据结构

这棵树的根结点位于足球比赛的现场，树叶为各个用户的终端，其他中转站为该树的内部结点。其中，方形结点代表转播站，圆形结点代表用户的终端。

从转播站到转播站以及从转播站到所有用户终端的信号传输费用都是已知的，一场转播的总费用等于传输信号的费用总和。

现在每个用户都准备了一笔费用想观看这场精彩的足球比赛，有线电视网有权决定给哪些用户提供信号。收费转播系统可以找到一个方案，使得有线电视网在不亏本的情况下使得观看转播的用户尽可能的多。

【例 1-3】交通咨询系统。

假若要在计算机上建立一个交通咨询系统，则可以采用图形结构或网状结构来表示实际的交通网络。如图 1-4 所示，图中顶点表示城市，边表示城市间的交通联系。

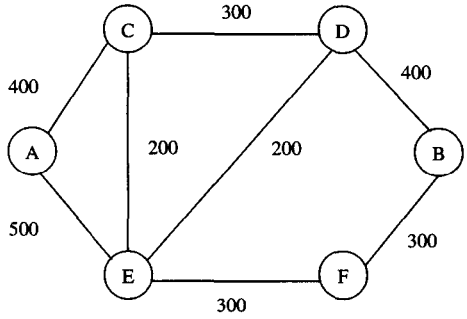

图 1-4　交通咨询系统的数据结构

这个咨询系统可以回答旅客提出的各种问题。例如，一位旅客要从 A 城市到 B 城市，他希望选择一条途中中转次数最少的路线。假设图中每一站都需要换车，则这个问题反映到图上就是要找一条从顶点 A 到顶点 B 所含边的数目最少的路径。我们只需从顶点 A 出发对图做广度优先搜索，一旦遇到顶点 B 就终止。由此所得的广度优先生成树上，从顶点 A 到顶点 B 的路径就是中转次数最少的路径，路径上 A 与 B 之间的顶点就是途径的中转站数，但是，这只是一类最简单的图的最短路径问题。有时，对于旅客来说，可能更关心的是节省交通费用；而对于司机来说，里程和速度则是他们感兴趣的信息。为了在图上表示有关信息，可对边赋

予权，权的值表示两城市间的距离，或行驶时间，或交通费用等等。此时，路径长度的度量就不再是路径上边的数目，而是路径上权值之和。

由以上 3 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式，而是诸如表、树、图之类的数据结构。因此，解决问题的一个关键步骤是选取合适的数据结构表示该问题，然后才能写出有效的算法。

学习数据结构的目的是为了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，通过算法训练来提高学生的思维能力，通过程序设计的技能训练来提高学生的综合应用能力和专业素质。

1.3 算法和算法分析

算法与数据结构关系紧密，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。下面就从算法特性、算法描述、算法性能分析与度量等 3 个方面对算法进行介绍。

1.3.1 算法及其特性

算法（Algorithm）是对特定问题求解步骤的一种描述。通俗地讲，一个算法就是一个解题方法。更严格地说，算法是由若干条指令组成的有限序列，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。一个算法必须总是（对任何合法的输入值）在执行有穷步之后结束，且每一步都可在有穷时间内完成。

（2）确定性。算法中的每一步必须有确切的含义，并且，在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的输出。

（3）可行性。一个算法是可行的，即算法中描述的运算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，它们是在算法开始前赋予算法的最初的数据，这些输入取自特定的数据集合。

（5）输出。一个算法具有一个或多个输出，这些输出是同输入之间存在某种特定关系的数据。

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，操作系统，只要整个系统不遭到破坏，它将永远不会停止，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机器可执行的，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的特定的实现。一个算法若用程序设计语言来描述，则它就是一个程序。

1.3.2 算法描述

算法可以使用各种不同的方法来描述。

最简单的方法是使用自然语言。用自然语言来描述算法的优点是简单且便于人们对算法的阅读。缺点是不够严谨。

算法还可以使用程序流程图、N-S 图等算法描述工具来描述，其特点是描述过程简洁、明了。使用以上两种方法描述的算法不能够直接在计算机上执行，若要将它转换成可执行的程序则还有一个编程的问题。

算法也可以直接使用某种程序设计语言来描述，不过直接使用程序设计语言并不容易，而且不太直观，常常需要借助于注释才能使人看明白。

为了解决理解与执行这两者之间的矛盾，人们常常使用一种称为伪代码语言的描述方法来进行算法描述。伪代码语言介于高级程序设计语言和自然语言之间，它忽略高级程序设计语言中一些严格的语法规则与描述细节，因此它比程序设计语言更容易描述和被人理解，而比自然语言更接近程序设计语言，它虽然不能直接执行，但很容易被转换成高级语言。本书中的算法绝大多数就是采用伪代码或自然语言来描述的。

1.3.3 算法性能分析与度量

解决同一个问题，往往能够编写出许多不同的算法。例如，对于排序问题，在第 9 章中将介绍多种算法。进行算法评价的目的，一方面在于从解决问题的不同算法中选择较为合适的一种，另一方面在于对现有算法的改进，从而设计出更好的算法。

究竟如何评价一个算法的好坏呢？显然，算法首先应该是正确的，此外，还要考虑以下 3 个因素：

（1）执行算法所耗费的时间；

（2）执行算法所耗费的存储空间，其中主要考虑辅助存储空间；

（3）算法应易于理解，易于编码，易于调试等。

一个好的算法应有效使用存储空间和有较高的时间效率，本节主要从一个算法的时间复杂度与空间复杂度来讨论算法的时间效率和空间效率。

一个算法转换成程序并在计算机上执行时，其运行所需要的时间取决于下列因素：

（1）硬件的速度。例如，使用奔腾 III 还是奔腾 4。

（2）书写程序的语言。实现语言的级别越高，其执行效率就越低。

（3）编译程序所生成目标代码的质量。对于代码优化较好的编译程序，其所生成的程序质量较高。

（4）问题的规模。即算法求解问题的输入量或初始数据量。例如，求 100 以内的素数与求 10000 以内的素数其执行时间必然是不同的。

显然，在各种因素都不能确定的情况下，很难比较出算法的执行时间。也就是说，使用执行算法的绝对时间来衡量算法的效率是不合适的。为此，可以将上述各种与计算机相关的软、硬件因素都确定下来，这样一个特定算法的运行工作量的大小就只依赖于问题的规模（通常用正整数 n 表示），或者说它是问题规模的函数。

1. 时间复杂度

一个算法所耗费的时间，是指该算法中每条语句的执行时间总和，而每条语句的执行时间是该语句的执行次数（也称为语句频度）与该语句执行一次所需时间的乘积。设执行每条语句所需的时间均为单位时间，则一个算法的时间耗费就是该算法中所有语句的频度之和。

一般情况下，算法中频度最大的语句执行次数是问题规模 n 的某个函数 f(n)，算法的时间量度记作

$$T(n) = O(f(n))$$

它表示当问题规模 n 趋向无穷大时，算法执行时间的增长率和 f(n)的增长率相同，称为算法的渐近时间复杂度，简称时间复杂度（Time Complexity）。例如，设一个算法中频度最大的语句执行次数为 $f(n) = n^2+8n+5$，则 $T(n) = 0(n^2)$。

下面举例说明如何求算法的时间复杂度。

【例 1-4】求两个 n 阶方阵的乘积 C=A*B，算法如下：

```
#define  n 30                    /*矩阵的最大阶数*/
void  Mult(int a[][n],int b[][n],int c[][n])   /*C=A*B*/
{ int i,j,k;
  for(i=1;i<=n;++i)
    for(j=1;j<=n;++j)
        {  c[i][j]=0;
           for(k=1;k<=n;++k)
                 c[i][j]+=a[i][k]*b[k][j];
        }
}
```

该算法中，频度最大的语句是“c[i][j]+ = a[i][k]*b[k][j];”，其频度为 $f(n) = n^3$，所以该算法的时间复杂度为 $T(n) = O(n^3)$。由此可见，当有若干个循环语句时，算法的时间复杂度是由嵌套层数最多的循环语句中最内层语句的频度 f(n)决定的。

有些情况下，很多算法的时间复杂度不仅仅是问题规模 n 的函数，还与它所处理的数据集的状态有关。这时，通常是根据数据集可能出现的最好情况和最坏情况，估计出算法的最好时间复杂度和最坏时间复杂度。有时，我们也对数据集的分布作出某种假设（如等概率），并讨论算法的平均时间复杂度。

【例 1-5】设有一组数据，对其进行冒泡排序，排序结果使得该组数据由小到大排列。冒泡排序的算法如下：

```
void Bubble_sort(int a[],int n)
{/*将 a 数组中的整数序列按从小到大的顺序排列*/
  int i,j;
  int change,temp;
  for(i=n-1,change=1;i>=1&&change;--i)
   { change=0;
     for(j=0;j<i;++j)
      if(a[j]>a[j+1])
        { temp=a[j];
          a[j]=a[j+1];
          a[j+1]=temp;
          change=1;
        }
   }
}/*bubble_sort*/
```

该算法的基本操作是比较或交换序列中相邻的两个整数，当 a 中初始序列为从小到大有序时，排序过程中的比较次数为 n-1，交换次数为 0；当初始序列为从大到小有序时，排序过程中的比较次数为 n(n-1)/2，交换次数为 3n(n-1)/2。因此，冒泡排序算法的最好时间复杂度

为 O(n)，最坏时间复杂度为 $O(n^2)$。除特别指明外，本书中涉及到的时间复杂度均指最坏时间复杂度。

另外，当一个算法的执行时间是一个与问题规模 n 无关的常数时，该算法的时间复杂度为常数阶，记作 T(n) = O(1)。

常见的时间复杂度，按数量级递增排列为：

$$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<O(2^n)$$

2. 空间复杂度

一个算法的空间复杂度（Space Complexity），记作 S(n)，是指该算法在运行过程中所耗费的辅助存储空间，它也是问题规模 n 的函数。若一个算法所耗费的辅助存储空间与问题规模 n 无关，记作 S(n) = O(1)。

算法所耗费的存储空间包括 3 部分，即算法本身所占用的存储空间、算法的输入输出所占用的存储空间和算法在运行过程中临时占用的辅助存储空间。其中，算法的输入输出所占用的存储空间是由算法要解决的问题规模决定的，它不随算法的不同而改变。算法本身所占用的存储空间与实现算法的程序源代码长短有关，要压缩这部分的存储空间，就必须编写较短的算法。算法在运行过程中临时占用的辅助存储空间随算法的不同而异，有的算法只需要占用少量的临时工作单元，而且不随问题规模的大小而改变；有的算法需要占用的临时工作单元数与解决问题的规模 n 有关，它随着 n 的增大而增大。

例如，前面讨论的冒泡排序算法，在运行过程中临时占用的工作单元数与解决问题的规模 n 无关，故该算法的空间复杂度为 O(1)；而在第 9 章讨论的二路归并排序算法的空间复杂度为 O(n)。

另外，若一个算法是递归算法，那么其空间复杂度与递归所需的栈空间的大小有关。例如，在第 9 章讨论的快速排序算法的最好空间复杂度为 $O(\log_2 n)$，最坏空间复杂度为 O(n)。

小　结

本章简要地介绍了数据、数据结构等基本概念；阐述了数据结构所包含的 3 个方面的内容，即数据的逻辑结构、数据的存储结构和数据的运算；讨论了线性结构和非线性结构的逻辑特征，以及数据存储的 4 种基本方法。希望读者学习这些内容和例子后，能对数据结构的基本概念具有初步的认识和理解。

算法和数据结构密切相关，不可分割，本章引进了算法、算法的时间复杂度等概念，以及分析时间复杂度的简易方法。

习　题

一、选择题

1. 算法的时间复杂度取决于（　　）。

A）问题的规模　　B）待处理数据的初态　　C）A 和 B

2. 计算机算法指的是解决问题的步骤序列，它必须具备（　　）这 3 个特性。

A）可行性、可移植性、可扩充性
B）可行性、确定性、有穷性
C）确定性、有穷性、稳定性
D）易读性、稳定性、安全性

3．下面关于算法描述错误的是（　　）。
A）算法最终必须由计算机程序实现
B）为解决某问题的算法与为该问题编写的程序含义是相同的
C）算法的可行性指的是指令不能有二义性
D）以上几个都是错误的

4．下面说法错误的是（　　）。
（1）算法原地工作的含义是指不需要任何额外的辅助空间
（2）在相同的规模 n 下，复杂度 O(n)的算法在时间上总是优于复杂度 O(2n)的算法
（3）所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界
（4）同一个算法，实现语言的级别越高，执行效率就越低
A）（1）　　B）（1）和（2）
C）（1）和（4）　　D）（3）

5．从逻辑上可以把数据结构分为（　　）两大类。
A）动态结构和静态结构　　B）顺序结构和链式结构
C）线性结构和非线性结构　　D）初等结构和构造型结构

6．在下面的程序段中，对 x 赋值的语句频度为（　　）。

```
for(i=0;i<n;i++)
      for(j=0;j<n;j++)
x=x+1;
```

A）O(2n)　　B）O(n)　　C）$O(n^2)$　　D）$O(\log_2 n)$

7．下面的程序段中，n 为正整数，则最后一行的语句频度，在最坏情况下是（　　）。

```
for(i=n-1;i>=1;i--)
    for(j=1;j<=i;j++)
      if (a[j]>a[j+1])
        a[j]与a[j+1]对换
```

A）O(n)　　B）O(nlog2n)　　C）$O(n^3)$　　D）$O(n^2)$

二、填空题

1．数据的物理结构包括_____的表示和_____的表示。

2．对于给定的 n 个元素，可以构造出的逻辑结构有_____、_____、_____和_____4 种。

3．数据结构中评价算法的两个重要指标是_____。

4．数据结构是研讨数据的_____和_____，以及它们之间的相互关系，并对这种结构定义相应的_____，设计出相应的_____。

5．一个算法具有 5 个特性：_____、_____、_____、有零个或多个输入

和有一个或多个输出。

6. 已知如下程序段：

```
for(i=n;i>0;i--)                    //语句1
{  x=x+1;                           //语句2
   for(j=n;j>=i;j--)                //语句3
     y=y+1;                         //语句4
}
```

语句 1 执行的频度为_____；语句 2 执行的频度为_____；语句 3 执行的频度为_____；语句 4 执行的频度为_____。

7. 在下面的程序段中，对 x 赋值的语句频度为_____（表示为 n 的函数）。

```
for(i=0;i>n;i++)
 for(j=0;j>i;j++)
   for(k=0;k>j;k++)
     x=x+delta;
```

8. 下面程序段中，带下划线的语句执行次数的数量级是_____。

```
i=1; while(i<n) i=i*2;
```

9. 执行下面的语句时，语句 s 的执行次数为_____。

```
for(i=1;i<n-1;i++)
  for(j=n;j>=i;j--)    s;
```

三、判断题

1. 数据元素是数据的最小单位。（　　）
2. 数据的逻辑结构是指数据的各数据项之间的逻辑关系。（　　）
3. 数据的逻辑结构说明数据元素之间的关系，它依赖于计算机的存储结构。（　　）
4. 健壮的算法不会因非法的输入数据而出现莫名其妙的状态。（　　）
5. 数据的物理结构是指数据在计算机内的实际存储形式。（　　）
6. 在顺序存储结构中，有时也需要存储数据元素之间的关系。（　　）

四、应用题

1. 数据元素之间的关系在计算机中有几种表示方法？各有什么特点？
2. 评价一个好的算法，是从哪几方面来考虑的？
3. 解释和比较以下概念：

（1）算法的时间复杂度

（2）算法

（3）语句频度

4. 对于一个数据结构，一般包括哪 3 个方面的讨论？
5. 在编制管理通讯录的程序时，什么样的数据结构较为合适？为什么？
6. 若有 100 个学生，每个学生有学号、姓名和平均成绩，若一般无增删操作，则采用什么样的数据结构最方便？写出这些结构。