



普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

SHUJU JIEGOU

数据结构

主 编 胡大威
副主编 陈兴无 安丰彩
主 审 蒋川群



中国电力出版社
www.infopower.com.cn

十一五

普通高等教育“十一五”国家级规划教材 (高职高专教育)
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

SHUJU JIEGOU
数据结构

主 编 胡大威
副主编 陈兴无 安丰彩
主 审 蒋川群



中国电力出版社

www.infopower.com.cn

Computer Technology

内容提要

本书为普通高等教育“十一五”国家级规划教材。本书系统地介绍了各种典型数据结构的基本概念、存储结构及其各种运算的原理和算法。全书共分10章,包括线性表、栈和队列、串、数组和广义表、树、图、查找、内外部排序和文件等方面的内容,在书的最后给出了实习题和一个综合应用实例。本书内容安排适当,浅显易懂。各章的“基本内容”和“学习要求”可以引导读者抓住重点。书中算法丰富,且采用C语言描述,并给出了部分完整程序供读者上机时参考。各章后的习题题型多样、题量丰富、难易恰当,便于读者理解和掌握课程内容。本书可作高等职业技术学院计算机及其相关专业的教材,也适合作大、中专院校和职业学校等有关专业的教材,还可供计算机应用技术人员及自学者使用。

图书在版编目(CIP)数据

数据结构 / 胡大威主编. —北京: 中国电力出版社, 2007.9

普通高等教育“十一五”国家级规划教材. 高职高专教育

ISBN 978-7-5083-5685-3

I. 数… II. 胡… 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(2007)第111119号

丛 书 名: 普通高等教育“十一五”国家级规划教材(高职高专教育)

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

服务电话: (010) 58383411

传 真: (010) 58383267

E-mail: infopower@cepp.com.cn

印 刷: 北京市同江印刷厂

开本尺寸: 185mm×260mm 印 张: 14.25 字 数: 330千字

书 号: ISBN 978-7-5083-5685-3

版 次: 2007年9月北京第1版

印 次: 2007年9月第1次印刷

印 数: 0001—4000册

定 价: 22.00元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

“数据结构”是介于数学、计算机软件 and 计算机硬件之间的一门计算机学科的核心专业课程，主要研究如何合理地组织大量数据，如何在计算机中更有效地存储和处理数据，以及如何设计出正确而高效的算法。数据结构是计算机程序设计、数据库、操作系统、编译原理及人工智能等领域的重要基础，同时也广泛地应用于信息学、系统工程等各种领域。数据结构不仅为程序设计提供了方法论性质的指导，还在更高层次上总结了现实生活中多种数据的计算机处理方法。数据结构是一门实用性很强的课程，学好它不但要在理论上打好基础，更要在实践上多下功夫。

随着计算机加工处理的对象从简单的数值发展到一般的符号，进而发展到具有一定结构的数据，数据结构也相应地由简单变得越来越复杂。本书介绍各种常用数据结构的基本概念和逻辑特征，讨论它们在计算机中的存储表示，定义了在这些数据结构上的运算，并对算法的效率进行了简要的分析。

全书共分 10 章，第 1 章介绍了数据结构和算法的基本概念；第 2~4 章介绍了线性表、栈和队列及串等几种基本的线性结构；第 5 章介绍了数组和广义表；第 6、7 章介绍了树和图这两种非线性结构；第 8、9 章介绍了数据处理中广泛使用的技术——查找和内排序；第 10 章对外部排序和文件作了简要的介绍。以上各章均附有一定数量的习题。在书的最后给出了实习指导、实习题和一个数据结构的应用实例——迷宫问题，以便读者上机实习。

本书可作为高职高专院校计算机及其相关专业的教材，还可供广大数据处理工作者、计算机应用技术人员及自学者使用。本书内容取舍适当，浅显易懂，既注重原理又重视实践，加大了算法实现和实习的分量，每章前面的“基本内容”和“学习要求”可以引导读者抓住重点。书中算法丰富，大部分采用 C 语言描述成可直接上机执行的程序，并作了较详细的注释，有利于读者理解算法的实质内容和基本思想。各章的习题题型多样、题量丰富、难易恰当，便于学生理解和掌握课程内容。各章中打星号的部分可以酌情作为选修内容。

本书第 1、2、3、4、6 章及实习部分由胡大威编写，第 5、7、8 章由陈兴无编写，第 9、10 章及综合应用实例由安丰彩编写，全书由胡大威统稿。

限于编者水平和经验，书中疏漏之处在所难免，惟望广大读者不吝赐教。

编 者

2006 年 11 月于武汉

目 录

前 言

第 1 章 绪论	1
1.1 基本术语和基本运算	2
1.2 算法描述和算法分析	5
习题 1	9
第 2 章 线性表	12
2.1 线性表	12
2.2 线性表的顺序存储结构	14
2.3 线性表的链式存储结构	19
习题 2	33
第 3 章 栈和队列	36
3.1 栈	36
3.2 栈的存储结构	37
3.3 队列	41
3.4 队列的存储结构	42
习题 3	48
第 4 章 串	51
4.1 串的基本概念	51
4.2 串的基本运算	52
4.3 串的存储结构	53
4.4 串的基本运算的实现	56
习题 4	58
第 5 章 数组和广义表	61
5.1 数组	61
* 5.2 矩阵的压缩存储	65
* 5.3 广义表	72
习题 5	77
第 6 章 树	79
6.1 树	79
6.2 二叉树	81
6.3 遍历二叉树	85
6.4 线索二叉树	94
6.5 树和森林	97

6.6 哈夫曼树及其应用	103
习题 6.....	106
第 7 章 图	110
7.1 图的基本概念.....	110
7.2 图的存储结构.....	114
7.3 图的遍历	119
7.4 最小生成树	123
* 7.5 关键路径	127
* 7.6 最短路径	131
习题 7.....	133
第 8 章 查找	136
8.1 基本概念	136
8.2 线性表的查找.....	137
8.3 二叉排序树查找.....	143
8.4 散列查找	148
习题 8.....	158
第 9 章 内部排序	161
9.1 基本概念	161
9.2 插入排序	163
9.3 交换排序	167
9.4 选择排序	174
9.5 归并排序	181
* 9.6 基数排序.....	183
9.7 各种排序方法比较.....	186
习题 9.....	188
*第 10 章 外部排序和文件	193
10.1 外部排序	193
10.2 文件的基本概念.....	200
10.3 顺序文件	202
10.4 索引文件	203
习题 10.....	204
实习部分	206
实习指导	206
实习题.....	208
综合应用实例——迷宫问题.....	215
参考文献	220

注：标有*的章节可作为选修内容；建议学时 64。

第 1 章 绪 论

“数据结构”是计算机科学与技术专业的一门专业基础课，它是介于数学、计算机软硬件和硬件三者之间的一门学科，主要研究如何合理地组织大量数据，如何在计算机中更有效地存储和处理数据，以及如何设计出正确而高效的算法。数据结构在计算机科学中有着十分重要的地位，它不仅是一般程序设计的基础，而且是编译原理、数据库系统、操作系统等专业课程的重要基础，并且还为软件开发和程序设计提供必要的技能训练。在众多的计算机系统软件和应用软件中都要用到各种数据结构。例如，操作系统中的存储管理、作业调度要用到链表及队列等结构；数组、链表及查找和排序算法在数据库技术中都得到了直接的应用；而文件的组织方式、文件的检索技术则是操作系统和数据库的重要组成部分。

在计算机发展的初期，计算机主要用于数值计算，所处理的对象是纯数值性的信息，程序设计者无需重视数据结构。随着计算机软、硬件的飞速发展和应用领域的不断扩大，计算机所处理的对象也从简单的纯数值信息逐步发展到字符、图形、图像、声音等各种复杂的、具有一定结构的非数值性数据，并且它们所占的比例越来越大。据统计，如今计算机 90% 以上的机器时间都是用来处理这类非数值性数据信息。新的计算机应用领域的不断扩大，要求人们不断探索新的数据表示形式，从而使得数据结构的内容得到进一步丰富和发展。

为了设计出效率高、可靠性好的程序，人们不但要掌握一般的程序设计方法，而且还必须研究计算机程序加工的对象，即研究各种数据的特性及数据之间的关系。数据是现实世界中的对象的某些特性和特征的部分抽象。针对每一种新的应用领域的处理对象，如何选择合适的表示形式，如何有效地组织计算机存储，在此基础上又如何有效地实现对象之间的运算关系，就是数据结构要研究和解决的问题。

基本内容

本章将介绍数据结构的基本概念和基本术语、算法及算法时间复杂度的分析方法。

学习要求

- (1) 熟悉各种基本术语、数据结构的逻辑结构与存储结构、数据结构的四种基本存储方法，特别要了解数据的逻辑结构与存储结构、数据结构与数据类型之间的区别。
- (2) 了解算法的性质、算法与程序的区别和联系、数据结构与算法之间的关系。
- (3) 掌握算法的书写规范。
- (4) 结合实例了解估算算法时间复杂度的方法，这是本章的难点。

1.1 基本术语和基本运算

1.1.1 基本术语

在大多数情况下, 计算机程序处理的信息(数据)之间往往具有重要的结构关系, 这就是数据结构研究的内容。例如电话号码查询系统。

设有一个电话号码簿, 它记录了 N 个人的名字和其相应的电话号码, 假定按如下形式安排: $(a_1, b_1) (a_2, b_2) \cdots (a_n, b_n)$ 其中 $a_i, b_i (i=1, 2, \cdots, n)$ 分别表示某人的名字和对应的电话号码。要求设计一个算法, 当给定任何一个人的名字时, 该算法能够打印出此人的电话号码, 如果该电话簿中没有此人, 则该算法能够报告查无此人。

上述的问题是一种数据结构问题。算法的设计, 依赖于计算机如何存储人的名字和对应的电话号码, 或者说依赖于名字和其电话号码的结构。此例中, 可将名字和对应的电话号码设计成二维数组、表结构、向量。而采用不同的数据结构, 将直接影响算法的选择和效率。

1. 数据

数据是承载信息的物理符号。描述现实对象的数、字符, 以及所有能输入到计算机中进行加工处理的信息统称为数据, 它是计算机程序加工的“原料”, 其含义极为广泛。

2. 数据元素

数据元素是数据的基本单位。它可以是单一数据项, 也可以是若干逻辑上有联系的相关数据项的组合, 数据项是数据的不可分割的最小单位。

数据元素视不同场合有其他同义语, 在顺序结构中称为元素, 在链式结构中称为结点, 在图中称为顶点, 在文件中称为记录等。

3. 数据对象

数据对象是具有相同性质的数据元素的集合, 是现实世界实体对象的数据抽象, 是数据的一个子集。例如, 为了描述数据化的“学生”这个实体对象, 假定姓名、学号、年龄、性别、系等这些数据项是你所关心的, 那么, 某一具体的学生“张三”、学号“995341”、年龄“20”、性别“男”、系“计算机系”就是构成数据对象“学生”的一个数据元素。

4. 数据结构

1) 数据结构的概念

数据对象中的数据不是孤立的, 而是彼此相关的, 数据元素相互之间的关系称为结构。根据数据元素之间关系的不同特征, 结构可分为集合、线性结构、树形结构、图状与网状结构四种基本类型。

将相互间存在一种或多种特定关系的数据元素的集合称为**数据结构**, 即数据结构是带有结构的元素的集合。虽然至今尚未有一个关于数据结构的定义, 但直观上可以理解为: 数据结构是一门研究在程序设计中计算机操作的对象, 以及它们之间的关系和运算的学科。它包含了三个方面的内容: 它研究了计算机需要处理的数据对象和对象之间的关系;

它刻画了应用中数据的逻辑组织；它描述了数据在计算机中存储、传送和转换的方式。

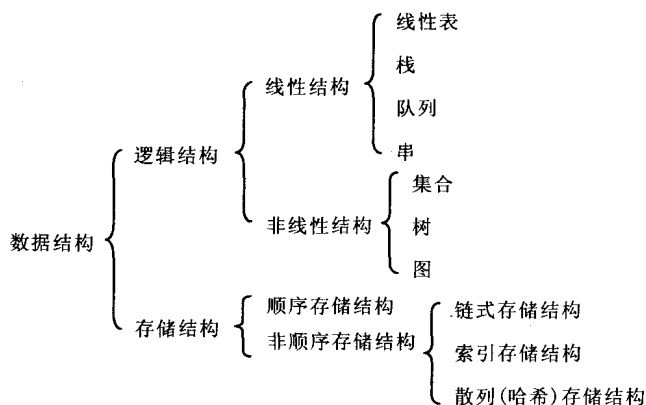
数据结构研究的是数据元素之间的逻辑关系(逻辑结构)和这种关系在计算机中的存储表示(存储结构)，并对每种结构定义各自的运算和设计出相应的算法。

数据结构不仅包含数据的逻辑结构和数据的存储结构，而且还包含数据的运算，所以应该将数据的逻辑结构、数据的存储结构和数据的运算看成一个整体。

数据结构和数据对象是不同的，在描述一种数据结构时，不但要描述数据对象，还要描述数据元素之间的关系。

2) 数据结构的分类

数据结构按其内容可作如下分类：



线性结构的逻辑特征是：如果结构是非空集，则有且仅有一个开始结点和一个终端结点，且每一个结点最多只有一个直接前趋和一个直接后继。

非线性结构的逻辑特征是：一个结点可能有多个直接前趋和直接后继。

顺序存储结构是把逻辑上相邻的结点存储在物理上相邻的存储单元中，结点的逻辑关系由存储单元的邻接关系来体现。其特点是借助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。通常用高级语言中的数组类型来描述顺序存储结构。

链式存储结构是指结点间的逻辑关系由结点的指针来表示的存储结构，逻辑上相邻的结点不一定存储在物理上相邻的存储单元中。其特点是借助指示元素存储地址的指针来表示数据元素之间的逻辑关系。通常用高级语言中的指针类型来描述链式存储结构。

索引存储结构是在存储结点数据的同时，还建立索引表，通过索引表来查找结点所在的存储位置的一种存储结构。

散列存储结构(也称为哈希存储结构)是根据结点关键字直接计算出结点的存储地址的一种存储表示。

各数据元素之间的逻辑关系是用户按使用需要建立起来、并呈现在用户面前的数据元素的结构形式，这种形式称为数据的逻辑结构。在不易混淆的情况下，常常将数据的逻辑结构简称为数据结构。

数据结构在计算机内的实际存储形式称为数据的存储结构，也称为数据的物理结构。它是逻辑结构到计算机存储器的映射。

数据的逻辑结构和存储结构是密切相关的，二者可能一致，也可能不一致。在有的结构类型中，如一维数组中，这二者是一致的。在有的结构类型中，如线性表中，这二者则可能是不一致的。在线性表中数据的逻辑关系表现为数据元素的有序性，即除了第一个和最后一个元素外，每个中间元素均有确定的前驱和后继。其存储结构若是顺序的，则同逻辑结构保持一致；若是链接的，则同逻辑结构不一致。另外，即使是线性表，若其元素的插入、删除运算被限制在表的一端进行，则该线性表便成为栈结构；若限制元素插入在表的一端、删除在表的另一端进行，则该线性表便成为队列结构。由此可见，数据的逻辑结构和运算方式也密切相关。

5. 数据类型

数据类型是和数据结构密切相关的一个概念，它最早出现在高级程序设计语言中，用来刻画程序操作对象的特性。数据类型是程序设计语言中已经实现了的数据结构，它规定了数据的取值范围、存取方式及允许进行的运算。数据类型的引进基于数据的两个特征：一是数据有值，并取值于一定的范围内；二是对值可以施行给定的运算。数据的这两个特征用来区分不同种类的数据，因此把数据类型定义为对数据对象值域和关于值域可施行的运算的一种分类。类型不同，值的存储表示、值的解释及值的运算方式也不同。如 C 语言中的 `int` 型变量一般占用两个字节，可进行算术、关系运算等。

程序设计语言允许用户直接使用的数据类型由具体语言决定。数据类型反映了程序设计语言的数据描述和处理能力。C 语言除了提供整型、实型、字符型等基本类型数据外，还允许用户自定义各种类型数据，例如数组、结构和指针等。

数据结构是在程序设计语言的基础上由用户建立起来的，它依靠程序设计语言提供的数据类型来描述数据的逻辑结构，也依靠程序设计语言提供的各种设施来定义、描述运算及算法。这些运算可根据用户实际需要自己定义，而不是由程序设计语言系统事先规定。因此，数据类型和数据结构的主要区别是：前者面向系统；后者面向对象，是高一层次的数据抽象。

1.1.2 基本运算

数据结构必须给出每种结构类型所定义的各种运算的相应算法。数据结构的基本操作有如下几种：

- (1) 建立与消除：建立和消除一个数据结构。
- (2) 访问：访问数据结构中的某个数据元素。
- (3) 插入：在数据结构中的指定位置上增添新的数据元素。
- (4) 删除：删除数据结构中指定的某个数据元素。
- (5) 修改：修改数据结构中指定的某个数据元素的值。
- (6) 查找：在数据结构中寻找满足某个特定要求的数据元素的值。
- (7) 排序：重新安排数据元素之间的逻辑顺序关系，使其值按由小到大或由大到小的次序排列。

为了增加对数据结构的感性认识，下面举一个例子来具体说明。

假设要在某大学的全体学生中查找一位叫“张三”的学生的情况，要求设计一个算法，

若找到,则给出姓名、学号、班级和系别;若没有找到,则显示“查无此人”。

通过分析可以发现,这个算法的设计依赖于学生的有关数据的逻辑结构及在计算机内的存储方式。在这个问题中,数据元素是学生,全校学生构成数据对象,全校学生按不同的方式组织起来就形成了不同的逻辑结构,查找的方法相应也会有所不同。

如果学生的姓名是随意排列毫无规律的话,则只能从学生名单中第一个姓名开始,逐个与“张三”比较,直到找到为止;若查完整个名单还没有找到此人,则输出“查无此人”。这种组织方法采用了线性结构,相应的查找方法是顺序查询,这种查找方法很简单,却相当费时,效率太低。

如果将学生的姓名进行适当的组织,将它们按汉语拼音字母的顺序排列起来,并且建立一个索引表,就可先从索引表中找到以字母“Z”开头的姓名在学生名单中的起始位置,然后从该处开始往下查找,而不用去查找其余 25 个字母开头的姓名。由于采用了这种索引结构来组织数据,相应地就使用了另一种完全不同的查找方法——索引查找,显然它的查找效率比前一种高。

如果将学生的名单按系别进行划分,每个系又按专业再细分,每个专业又按班级组织起来,则只要知道了张三所在的系别、专业和班级就很容易查清是否有此人。这里采用了树形结构来组织数据,相应地就可以使用树表的查询方法来查找。

学生名单的组织方式就是一个数据结构问题,三种不同的数据结构,可以得出完全不同的三种查找算法。由此可见,算法与数据结构密切相关,每一种算法依赖于具体的数据结构,数据结构直接影响着算法的选择和效率。

1.2 算法描述和算法分析

同一个问题的数据集合可以组织成不同的数据结构。例如,学生成绩档案在逻辑上属于线性数据结构,可以组织成线性表,也可以组织成二叉树。具体选择哪种结构,要视具体情况而定。此外,同为线性表,在存储时既可以选择顺序存储,也可以选择链式存储。由于数据的运算是通过算法来描述的,因此就产生了一个对数据结构和算法如何选择和评价的问题。讨论算法也就成了数据结构课程的重要内容之一。

1.2.1 算法

1. 定义

算法是指令的一个有限集合,它精确地描述了解题方法,即它是执行特定计算任务的有穷过程。

算法具有如下性质:

- (1) 动态有穷性。算法中每条指令的执行次数必须是有限的,最后一定要终止。
- (2) 确定性。算法中每条指令的含义都必须是明确的、无二义性的。
- (3) 输入。一个算法可以有 0 个或 0 个以上的输入,这些输入取自于某个特定对象的集合。

(4)输出。一个算法至少产生一个输出，这些输出是同输入有着某些特定关系的量。

(5)可行性。每条指令的执行时间都是有限的。

2. 算法与程序的区别

算法经过有限步计算后一定会终止，而程序不一定能满足动态有穷性。例如，操作系统是程序而不是算法，因为只要系统不遭到破坏，它永远也不会终止。此外，程序一般是用某种计算机语言书写的，而算法既可以采用计算机语言来描述又可以采用自然语言来描述。

3. 算法与数据结构的关系

算法和数据结构密切相关，同一应用问题采用不同的数据结构，其运算的算法也不相同。

著名的瑞士计算机科学家 N.Wirth 教授指出：算法+数据结构=程序，这里的算法指的是对数据运算顺序的描述，数据结构是指数据的逻辑结构和存储结构。程序设计的实质是对确定的问题选择一种好的数据结构，从而设计出一种好的算法。而好的算法在很大程度上取决于所采用的数据结构。

在学习本课程的过程中，不能仅仅以简单地实现一个算法为目标，而应注重从数据结构的角度来分析和解决问题，通过比较来合理地使用数据结构与算法，尽可能地做到让数据结构与算法完美结合。

4. 算法设计要求

(1)正确性：设计出的算法应当能满足具体问题的要求，至少应包括对于输入、输出和加工处理等的正确描述。

(2)可读性：设计出的算法应当有助于读者对算法的理解。

(3)健壮性：当输入的数据非法时，算法能作出适当的反应。

(4)高效率与低存储量需求：高效率是指在同一问题的多个算法中，选择执行时间短的算法；低存储量需求是指在同一问题的多个算法中，选择所需存储空间最小的算法。

1.2.2 算法描述

1. 算法的描述方式

算法可以用自然语言、数学语言、伪代码来描述，也可以用流程图描述，还可以用高级程序设计语言描述。本书中的数据结构和算法是用 C 语言来描述的，为了描述的方便和使算法清晰简明，本书又不完全拘泥于 C 语言的细节，读者上机调试算法时，只要作简单的修改即可。

数据结构的表示和运算的算法均涉及到程序设计语言，算法必须借助于语言符号来表达，用程序设计语言书写的算法才能在计算机上执行。C 语言提供了丰富的数据类型，数组、指针、记录、文件等数据类型可以方便地用来表示数据的存储结构。C 语言的语言结构体现了结构化程序设计原则，可自然地应用自顶而下、逐步求精的程序开发方式，是一种很适合描述算法的程序设计语言。

2. 算法的书写格式

所有算法均以如下函数形式表示：

函数类型 函数名(参数表)

```
/*算法说明*/
{ 语句序列
}
```

为便于读者阅读和书写算法，应对算法做出如下规定。

(1)在算法说明(以注释的形式写在算法的前面)中，应该包含以下信息：

- ①指明算法的功能；
- ②指出形式参数表中各形式参数的含义；
- ③说明引用了哪些外部变量，它们的初值如何，以及应满足哪些限制条件；
- ④在必要的时候还应陈述所采用的存储结构。

(2)算法的输入和输出途经如下：

- ①通过标准函数 `scanf` 和 `printf` 实现；
- ②以算法中形式参数表里列出的形参作为输入、输出的媒介；
- ③通过外部变量隐式地传递信息。

(3)所有的算法都要写成函数形式。

(4)在不易理解的语句后面，加上一定的注释以提高算法的可读性。

1.2.3 算法分析

同一个问题可能有多个不同的算法，这就自然需要一个衡量算法优劣的标准，一般从以下两个方面考虑：

- (1)算法在计算机中运行时所耗费的时间。
- (2)运行算法所占存储量的大小，其中主要考虑程序运行时所需辅助存储量的大小。

算法分析就是分析算法的时间开销和空间开销，前者叫时间复杂度，后者叫空间复杂度。理想的算法应既少花费时间，又少花费空间。但算法的执行时间和执行算法所需要的存储空间往往是矛盾的，两者很难兼顾，所以应根据具体情况有所侧重。一般来讲，人们比较注重于效率(减少时间开销)，常以算法的执行时间作为主要的衡量标准，这是因为，计算机硬件的价格越来越便宜，使得存储空间的费用也不断下降。

一个程序在计算机上运行时所耗费的时间取决于下列因素：

- (1)程序运行时所需输入的数据总量；
- (2)对源程序进行编译时所需时间；
- (3)计算机执行每条指令所需时间；
- (4)程序中的指令重复执行的次数。

很难做到精确计算时间复杂度，通常的做法是讨论算法在最坏情况下的时间复杂度，即分析最坏情况以估算算法执行时间的一个上界，只进行数量级的比较。

估计时间复杂度时，常采用计算语句频度的方法。一个语句的频度是指一个语句重复执行的次数。

算法所要求解问题的输入量(或初始数据量)称为问题的规模，用 n 来表示，如线性表的长度、多项式的项数、矩阵的阶、图中的顶点数等。

程序中所有语句的频度之和 $f(n)$ 称为算法的时间复杂度，它是问题的规模 n 的函数，即

$$f(n) = f_1(n) + f_2(n) + \cdots + f_k(n) = C_k n^k + C_{k-1} n^{k-1} + \cdots + C_1 n + C_0 \quad (1-1)$$

算法的时间复杂度的数量级 $O(n^k)$ 称为算法的渐近时间复杂度，即在一般情况下，算法中基本操作重复执行的次数是问题规模 n 的某个函数 $f(n)$ ，算法的渐近时间复杂度记作

$$T(n) = O(f(n)) \quad (1-2)$$

它表示随问题规模 n 的增大，算法执行时间的增长率和 $f(n)$ 的增长率相同。

在进行算法分析时，常常对算法的时间复杂度和算法的渐近时间复杂度不加以区别，而经常将算法的渐近时间复杂度简称为算法的时间复杂度。

算法时间复杂度的读法：算法的执行时间复杂度等于大欧 (O) 的 n 的 k 次方。

算法时间复杂度优劣的判别准则：一般把随 n 增大而函数 $f(n)$ 增长速度较慢的算法视为较优的算法。增长率由小到大的排列顺序为： $O(1)$ 、 $O(\lg n)$ 、 $O(n)$ 、 $O(n \lg n)$ 、 $O(n^2)$ 、 $O(n^3)$ 、 $O(2^n)$ 。

【例 1-1】 设有程序段 (1) $x++$;

求其时间复杂度。

解 行号 每行执行的次数

(1) 1

$f(n) = 1$ ，所以算法的时间复杂度为 $O(1)$ 。

【例 1-2】 设有程序段 (1) $\text{for}(i=1; i \leq n; i++)$

(2) $x++$;

求其时间复杂度。

解 行号 每行执行的次数

(1) $n+1$

(2) n

$f(n) = (n+1) + n = 2n+1$ ，所以算法的时间复杂度为 $O(n)$ 。

【例 1-3】 设有程序段 (1) $\text{for}(i=1; i \leq n; i++)$

(2) $\text{for}(j=1; j \leq n; j++)$

(3) $x++$;

求其时间复杂度。

解 行号 每行执行的次数

(1) $n+1$

(2) $n(n+1)$

(3) n^2

$f(n) = (n+1) + n(n+1) + n^2 = 2n^2 + 2n + 1$ ，所以算法的时间复杂度为 $O(n^2)$ 。

【例 1-4】 设有程序段 (1) $\text{for}(i=1; i \leq n; i++)$

(2) $\text{for}(j=1; j \leq n; j++)$

(3) $\{c(i, j) = 0;$

(4) $\text{for}(k=1; k \leq n; k++)$

(5) $c(i, j) = c(i, j) + a(i, k) + b(k, j); \}$

求其时间复杂度。

解 行号 每行执行的次数

(1) $n+1$

(2) $n(n+1)$

(3) n^2

(4) $(n+1)n^2$

(5) n^3

$f(n) = 2n^3 + 3n^2 + 2n + 1$, 所以算法的时间复杂度为 $O(n^3)$ 。

习 题 1

一、单项选择题

1. 数据结构是一门研究非数值计算的程序设计问题中计算机的_____, 以及它们之间的_____和运算等的学科。

- (1) A. 操作对象 B. 计算方法 C. 逻辑存储 D. 数据映像
(2) A. 结构 B. 关系 C. 运算 D. 算法

2. 在数据结构中, 从逻辑上可以把数据结构分成_____。

- A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
C. 线性结构和非线性结构 D. 内部结构和外部结构

3. 算法分析的目的是_____, 算法分析的两个主要方面是_____。

- (1) A. 找出数据结构的合理性 B. 研究算法中的输入和输出的关系
C. 分析算法的效率以求改进 D. 分析算法的易懂性和文档性
(2) A. 空间复杂性和时间复杂性 B. 正确性和简明性
C. 可读性和文档性 D. 数据复杂性和程序复杂性

4. 计算机算法指的是_____, 它必须具备_____这三个特性。

- (1) A. 计算方法 B. 排序方法
C. 解决问题的步骤序列 D. 调度方法
(2) A. 可执行性、可移植性、可扩充性 B. 可执行性、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

5. 算法的计算量的大小称为计算的_____。

- A. 效率 B. 复杂性 C. 现实性 D. 难度

6. 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

二、填空题

1. 数据结构是研讨数据的_____和_____, 以及它们之间的相互关系, 并对与

这种结构定义相应的_____，设计出相应的_____。

2. 数据的逻辑结构是指_____。
3. 数据的物理结构包括_____的表示和_____的表示。
4. 数据结构中评价算法的两个重要指标是_____。

三、简答及应用题

1. 简述下列术语：数据、数据元素、数据对象、数据结构和数据类型。
2. 数据的逻辑结构有哪几种类型？
3. 什么是数据的存储结构，有哪几类存储结构？
4. 试说明数据结构与算法的关系。
5. 有哪几类常用的描述算法的方法？
6. 指出下列各算法的时间复杂度：

(1)

```
for(i=1;i<=n;i++)
    for (j=1;j<=i;j++)
        {x=x+1;
         s=s+x;}
```

(2)

```
i=1;
while (i<=n)
    {x=x+1;
     i=i+1;}
```

(3)

```
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
        {x++;
         for (k=1;k<=j;k++)
             s=s+x;
        }
```

(4)

```
prime(int n)
{
    i=2;
    while ((n mod i !=0) and (i<sqrt(n)))
        i=i+1;
    if (i>sqrt(n)) printf("n is a prime number! ");
    else printf("n is not a prime number!");
}
```

(5)

```
sum1(int n)
{
    p=1; sum=0;
    for (i=1;i<= n ;i++)
        {p=p*i; sum=sum+p;}
    return (sum);
}
```

(6)

```
sum2 (int n)
{
    sum=0;
    for (i=1;i<= n ;i++)
        {p=1;
         for (j=1;j<= i ;j++)
```

```
    sum=sum+p;}  
    return (sum);  
}
```

7. 用 C 语言描述下列每一个算法，并分别求出它们的时间复杂度。

- (1) 试写一个算法，自大到小依次输出顺序读入的三个数 x 、 y 和 z 的值。
- (2) 求一维实型数组 $a[n]$ 中的所有元素之乘积。
- (3) 从二维字符数组 $b[m][n]$ 中查找出最大元素所在的行、列下标。