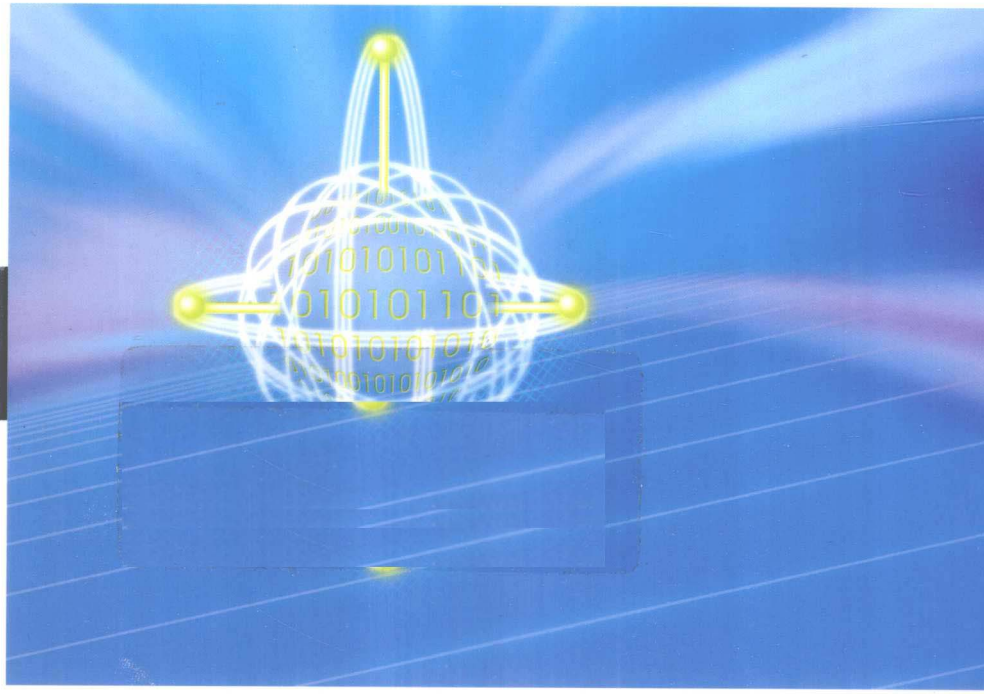
21世纪高等院校应用型规划教材

数据结构实用教程



提供电子教案

刘 波　郝振明　王晓明　编著

机械工业出版社
CHINA MACHINE PRESS

21 世纪高等院校应用型规划教材

数据结构实用教程

刘　波　郝振明　王晓明　编　著

机械工业出版社

本书系统地介绍了线性表、栈、队列、串、数组、广义表、树、图等常用的数据逻辑结构和存储结构，以及使用各种数据结构的基本操作、查找和排序算法等。各章以抽象数据类型、存储与表示、基本操作算法、应用实例、小结为线索组织相关内容，配有适量的练习题和上机操作题，不仅可以满足理论教学的需要，还可供读者用于理解知识及复习提高，并指导实验教学。

本书内容全面实用，概念清楚，体系合理，采用类C语言描述数据结构和操作算法，简明清晰、可读性好，容易转换成能够上机执行的C程序、C++程序或Java程序等。

本书可作为计算机类以及电子信息、管理信息系统、电子商务等相关专业的教材，也可供计算机科学与工程领域从业人员参考和查阅。

图书在版编目（CIP）数据

数据结构实用教程 / 刘波等编著. —北京：机械工业出版社，2009.12
（21世纪高等院校应用型规划教材）
ISBN 978-7-111-28938-8

Ⅰ.数… Ⅱ.刘… 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（2009）第198668号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
责任编辑：唐德凯
责任印制：杨 曦

北京鑫海金澳胶印有限公司印刷

2010年1月第1版·第1次印刷
184mm×260mm·18.5印张·454千字
0001—3000册
标准书号：ISBN 978-7-111-28938-8
定价：29.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：(010)88361066
销 售 一 部：(010)68326294
销 售 二 部：(010)88379649
读者服务部：(010)68993821

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

封面无防伪标均为盗版

前　言

数据是计算机处理的对象，而数据的组织与结构是计算机科学中最基本的研究内容。数据结构课程的主要内容就是研究数据在计算机中的表示方法、关联方法、存储组织方法以及在其上的操作处理方法。本书系统地介绍了软件设计中最常用的数据结构，如线性表、栈、队列、串、数组、广义表、树、图等；各种数据结构的逻辑关系和它们在计算机中的存储表示，以及在这些数据结构上的基本运算；另外，还介绍了查找和排序算法等。

数据结构是计算机类、电子信息、管理信息系统、电子商务等相关专业的必修课程，是后续专业课程和复杂程序设计的基础。它是一门理论性、实践性很强的课程，除要求学生掌握有关理论外，还要求掌握复杂程序设计的技能。本书以培养应用型本科人才为宗旨，重视理论与实践相结合，精选了较多应用实例，不仅有诸如最小路径、关键路径等典型问题，还有诸如操作系统调度等特色问题。通过应用举例和实例分析，学生能深刻理解算法的构造及用途，开阔思路，触类旁通，提高软件设计和编程能力，培养创新性思维和实践能力。

各章以抽象数据类型、存储与表示、基本操作算法、综合举例、小结为线索组织相关内容，章后配有难度适宜的习题（选择、填空、简答或应用、算法题）和上机实验题，不仅可以满足理论教学的需要，还可以帮助学生更好地理解课程内容，掌握算法设计所需的方法和技术，提高程序设计的技能，达到对所学知识融会贯通和灵活应用的目的。

全书采用类 C 语言描述数据结构和操作算法，简明清晰、可读性好，容易转换成能够上机执行的 C 程序、C++程序或 Java 程序等。

本书共分 10 章，内容精炼，是编者在讲授“数据结构”和“算法分析与设计”、“高级程序设计语言”等相关课程的教学经验的基础上，吸取各版本教材之精华，结合多年的备课教案和讲义编写而成的。其中的第 1、2、6 章由王晓明编写，第 4、5、7 章由刘波编写，第 3、8、9、10 章由郝振明编写，全书由刘波统稿。暨南大学研究生黎春桃及本科生吕烨、杨帆等为本书的插图制作做了大量工作，在此表示衷心的感谢。

本书可以作为高等学校计算机学科和信息类学科本科生和专科生教材，也可以作为其他理工科专业本科生的选修教材；对于企、事业单位从事信息类相关专业工作的科技工作者，也是一本实用的参考书。

本书最后附有习题参考答案，并免费提供电子教案，读者可从机械工业出版社网站 www.cmpedu.com 下载。由于作者水平有限，书中会存在一些不足之处，恳请读者赐教指正。

编　者

目　录

第1章 绪 论

本章主要介绍数据、数据元素、数据对象、存储结构和数据类型等基本概念和术语，以及抽象数据类型的定义、表示和实现方法，并且介绍了算法设计的基本要求和分析算法的方法等。要求在熟悉这些内容的基础上，重点理解算法 5 个要素的确切含义，掌握语句频度和估算算法时间复杂度的方法。

1.1 数据结构基本概念和术语

随着计算机应用领域的扩大和软、硬件的发展，计算机应用已不再局限于科学计算，而更多地用于控制、管理及数据处理等非数值计算的处理工作。解决“非数值计算问题”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。所以要想有效地使用计算机、充分发挥计算机的性能，还必须学习好数据结构的有关知识。所有的计算机系统软件和应用软件都要用到某些类型的数据结构。数据结构是计算机专业专业基础课，是介于数学、计算机硬件和计算机软件三者之间的一门十分重要的核心课程。它涉及在计算机中如何有效地表示数据、如何合理地组织数据和处理数据。还涉及初步的算法设计和算法性能分析技术。数据结构的主要内容包括数组、链接表、栈和队列、递归、树与森林、图、堆与排序、索引与散列结构等。课程采用类 C 语言作为算法的描述工具，强化数据结构基本知识和面向对象程序设计基本能力的双基训练。

1.1.1 什么是数据结构

用计算机解决实际问题，一般分下面几个步骤。

1）分析阶段：分析实际问题，从中抽象出一个数学模型。

2）设计阶段：设计出解决该数学模型的算法。

3）编程阶段：用适当的程序语言编写出执行的程序。

4）测试阶段：测试、修改直至得到解答。

数据结构就是要解决这 4 个步骤中第一阶段和第二阶段所提到的问题。对于数值计算问题，可以使用数学方程描述，如预报人口增长情况的数学模型为微分方程。但对于非数值计算问题，无法用数学方程描述，而是要设计出合适的数据结构，才能有效地解决问题，如学生信息检索系统。

表 1-1 中每个学生的情况为一个记录，它由学号、姓名、性别、专业和年级组成。

表 1-1 学生情况登记表

学号	姓名	性别	专业	年级
980001	刘立	男	计算机	2003 级
980002	王红	男	数学	2003 级
980003	李娜	女	经济	2004 级

用这张表建立了学生的姓名、专业、年级等相关信息的数学模型。解决这类数学模型的数据结构是一个线型的数据结构。

又如通信网络问题。在 n 个城市之间建立通信网络，要求在其中任意两个城市之间都有直接的或间接的通信线路。图 1-1 描述了 7 个城市之间的通信线路。其中，图中圆圈表示一个城市，两个圆圈之间的连线表示对应城市之间的通信线路，连线上的数值表示该通信线路的造价。

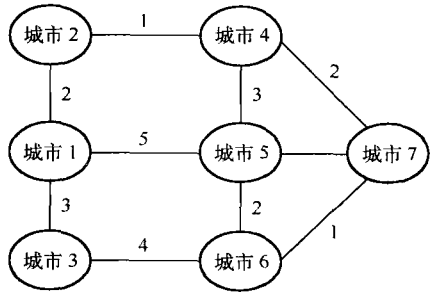


图 1-1　7 个城市的通信线路

通过以上这些例子可见，描述这类非数值计算问题的数学模型不再是数学方程，而是诸如图、表和树之类的数据结构。因此，数据结构是一门主要研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作的学科。

1.1.2　基本概念和术语的解释

在学习数据结构知识之前，先对一些基本概念和术语赋予确切的含义。

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能输入到计算机中并被计算机程序处理的符号的总称。如字符、图像、声音等计算机都能接受和处理，所以都是数据。

2．数据项（Data Item）

数据项是具有独立含义的标识单位，是数据不可分割的最小单位。如学生信息表中的姓名、学号、分数等。

3．数据元素（Date Element）

数据元素是数据的基本单位，通常作为一个整体进行考虑和处理。一个数据元素由若干个数据项组成。如学生信息表中的一个记录（一行数据）。

4．数据对象（Date Object）

数据对象是性质相同的数据元素的集合，是一个数据的子集。如在学生信息管理系统中，数据对象是全体学生。

5．数据结构（Date 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。数据结构可描述为：B=（D，R），其中 D 为有限个数据元素的集合，R 为有限个数据元素间关系的集合。根据数据元素之间关系的不同，通常有以下几种基本结构。

1）集合：结构中的元素除了同属于一个集合的关系外，别无其他关系。

2）线性结构：结构中的元素之间存在一对一的关系，有且仅有一个开始结点和终止结点，除开始结点外，每个结点有且仅有一个前趋结点，除终止结点外，每个结点有且仅有一个后续结点，如图 1-2 所示。

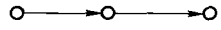


图 1-2　线性结构

3）非线性结构：数据元素之间不存在一对一的关系，可能有一个开始结点，但有多个终止结点。如树结构、图结构等。

树结构：结构中的元素之间存在一个对多个的关系，如图 1-3 所示。

图结构：结构中的元素之间存在多个对多个的关系，如图 1-4 所示。

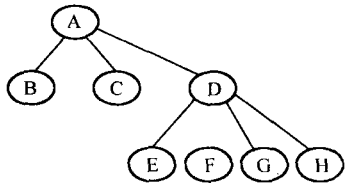


图 1-3　树结构

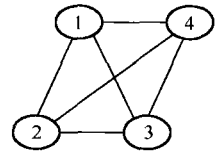


图 1-4　图结构

6．数据的逻辑结构（Logical Structure）

数据的逻辑结构是数据元素之间的相互关系的描述，它可以用一个数据元素的集合和定义在此集合上的若干关系来表示。数据的逻辑结构是从逻辑关系上来观察数据，它与数据的存储方式无关，是独立于计算机的。

7．数据的存储结构（物理结构）

数据结构在计算机中的表示（映像）包括数据结构中元素的表示及元素间关系的表示。数据的存储结构是数据的逻辑结构在计算机存储器里的实现，它是依赖计算机的。数据存储结构有顺序和链式两种不同的方式。

顺序存储结构：把逻辑上相邻的元素存储在物理位置相邻的存储单元里。一般用程序设计语言的数组来实现此存储结构。其特点是逻辑上相邻的元素其物理位置相邻。例如，L={元素 1，元素 2，…，元素 n} 的顺序存储结构如图 1-5 所示。

链式存储结构：用一组任意的存储单元存储结点数据，将每个结点所占的存储单元分为两个部分，一部分存放结点本身的信息，即数据项，另一部分存放该结点的后继结点所对应的存储单元的地址，即指针项。数据元素之间逻辑上的联系由指针来体现，其特点是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附设的指针字段来表示。例如，L={元素 1，元素 2，…，元素 n} 的链式存储结构如图 1-6 所示。

8．数据类型（Date Type）

数据类型就是一个值的集合和定义在这个值集上的一组操作的总称。例如，C 语言中的整型变量，其值集为某个区间上的整数（区间大小依赖于不同的机器），定义在其上的操作为加、减、乘、除和取模等算术运算。

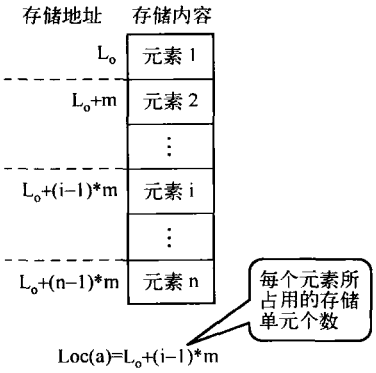
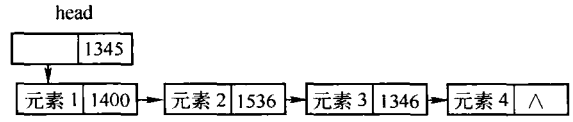


图 1-5　顺序存储结构



存储地址	存储内容	指针
1345	元素 1	1400
1346	元素 4	∧
⋮	⋮	⋮
1400	元素 2	1536
⋮	⋮	⋮
1536	元素 3	1346

图 1-6　链式存储结构

9. 抽象数据类型（Abstract Data Type，ADT）

抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义仅取决于它的逻辑特性，与它的存储方式无关。其格式定义：

```
ADT 抽象数据类型名{
    数据对象: <数据对象的定义>
    数据关系: <数据关系的定义>
    基本操作: <基本操作的定义>
} ADT 抽象数据类型名
```

其中，数据对象和数据关系的定义用伪代码描述，基本操作的定义格式为：

```
基本操作名(参数表)
    初始条件: <初始条件描述>
    操作结果: <操作结果的定义>
```

基本操作有两种参数，一是赋值参数，只为操作提供输入值；二是引用参数，以&打头，除可提供输入值外，还将返回操作结果。初始条件描述了操作之前数据结构和参数应满足的条件，若不满足，则操作失败，并返回相应出错信息。操作结果说明了操作正常完成之后，数据结构的变化状况和应返回的结果。若初始条件为空，则省略之。

例如，抽象数据类型三元组的定义：

```
ADT Triplet {
    数据对象:D= {e1,e2,e3 |e1,e2,e3∈ElemSet(定义了关系运算的某个集合)
    数据关系:R1={<e1,e2>,<e2,e3>}
    基本操作:
        InitTriplet (&T, v1, v2, v3)
        操作结果:构造了三元组 T,元素 e1,e2 和 e3 分别被赋以参数 v1,v2 和 v3 的值。
        DestroyTriplet (&T)
          操作结果：三元组 T 被销毁。
        Get (T, i , &e)
          初始条件：三元组 T 已存在，1≤i≤3 。
          操作结果：用 e 返回 T 的第 i 元 的值。
```

Put (&T, i, e)
初始条件：三元组 T 已存在，1≤i≤3 。
操作结果：改变 T 的第 i 元的值为 e 。
} ADT Triplet

1.2 算法和算法分析

算法与数据结构的关系紧密，在算法设计时要先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。下面对算法的特性、算法的描述、算法的设计、算法的度量等进行介绍。

1.2.1 算法的特性

算法（Algorithm）是对特定问题求解步骤的一种描述，是指令的有限序列。其中每条指令表示一个或多个操作。一个算法具有以下 5 个重要特性。

1）输入性：一个算法具有零个或若干可输入量，这些输入取自特定的数据对象集合。

2）输出性：一个算法至少产生一个输出或执行一个有意义的操作，这些输出与输入之间存在某种特定的关系。

3）有穷性：一个算法必须在有穷步之后结束，即必须在有限时间内完成。

4）确定性：算法的每一条指令的含义明确，无二义性，并且在任何条件下，算法只有唯一的一条执行路径。

5）可执行性：算法中的每一步都可以通过已经实现的基本运算的有限执行得以实现。

算法的含义与程序十分相似，但是与程序是有区别的。

1）一个程序不一定满足有穷性。

2）程序中的指令必须是机器可以执行的，而算法中的指令没有此要求。

3）一个算法，若用机器可执行的语言书写，则它就是一个程序了。

4）一个程序如果对任何输入都不会陷入无限循环，则它就是一个算法。例如，操作系统是一个程序，而不是一个算法。然而，可以把操作系统的各种任务看成是单独的问题，每个问题由一部分操作系统程序通过特定的算法实现，得到输出结果后便终止。

1.2.2 算法的描述

一个算法可以用多种描述方法，大致的描述方法有如下几种。

1）自然语言描述：使用人们日常使用的语言，如中文或英文等语言描述算法。其优点是简单且便于人们对算法的阅读，缺点是不够严谨，文字冗长，容易出现“歧义性”。

2）流程图或 N-S 图描述：流程图用一些图框表示各种操作，用带箭头的流程线反映流程的执行先后次序；N-S 图是完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他的从属于它的框，或者说，由一些基本的框组成一个大的框。流程图或 N-S 图描述的优点是直观、形象、易于理解，但画起来比较麻烦。因此，流程图或 N-S 图适宜表示一个算法，但在设计算法过程中使用不是很理想。

3）计算机语言描述：使用某一种计算机语言描述算法。其优点是可以在计算机上直接运行

并得到结果，缺点是直接使用计算机语言描述不太直观，常常需要借助于注释才能让人看明白。

4）伪代码描述：是用介于自然语言和计算机语言之间的文字和符号来描述算法，即每一行（或几行）表示一个基本操作。其优点是书写方便、自由、格式紧凑、比较好懂、容易表达出设计者的思想、便于向计算机语言算法（即程序）过渡。但用伪代码写算法不如流程图或N-S图直观，可能会出现逻辑上的错误。例如，循环或选择结构的范围搞错等。

以上介绍了常用的表示算法的几种方法，在程序设计中读者可以根据需要和习惯任意选用。

本书采用类C语言作为算法描述工具。类C语言是介于伪代码和C语言之间，用伪代码描述一些抽象算法，同时精选了C语言的一个核心子集，并作了若干扩充修改，增强了语言的描述功能。这使得数据结构与算法的描述和讨论简明，不拘泥于C语言的细节，又能容易转换成C或者C++程序。大多数存储结构定义中用ElemType表示任一数据类型。

1.2.3 算法的设计

设计一个好的算法应满足以下要求。

1）正确性：算法应该满足具体问题的要求，正确反映求解问题对输入、输出和加工等方面的需求。

2）可读性：算法应该便于人阅读与交流。可读性好有助于人对算法的理解，便于算法的交流与推广。

3）健壮性：当输入数据非法时，算法也能适当地作出反应或进行处理，而不会产生莫明其妙的输出结果。

4）效率和存储量：算法必须满足效率高，而且在给定的存储空间内正常运行。对于同一问题的多个算法，尽量选用效率高，占用存储空间少的算法较好。

1.2.4 算法的度量

在算法是正确的情况下，评价一个算法的优劣主要有两个指标，即时间复杂度和空间复杂度。

1．算法的时间复杂度

一个算法的时间复杂度（Time Complexity）是指算法编制完成程序后，在计算机中运行所消耗的时间。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)，算法的时间度量记做：

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

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，称做算法的渐近时间复杂度，简称时间复杂度。

在一个算法中，是按算法中重复执行次数最多的语句来确定算法的时间复杂度。语句频度（Frequency Count）指该语句在算法中重复执行的次数。

时间复杂度一般用数量级来表示。常见的时间复杂度是：

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

其中，$O(1)$为常量阶，$O(\log_2 n)$为对数阶，$O(n)$为线性阶，$O(n^2)$为平方阶，$O(2^n)$为指数阶。尽可能选用低阶多项式的算法，而不希望用指数阶的算法。

2．算法的空间复杂度

算法的空间复杂度(Space Complexity)是指程序运行从开始到结束所需的存储量，记做：

$$S(n)=O(g(n))$$

其中，g(n)为问题规模 n 的某个函数，S(n)被称为算法的渐近的空间复杂度，简称空间复杂度。

在一个算法中，是按程序运行所需的存储空间来确定算法的空间复杂度的。程序运行所需的存储空间包括以下两部分。

1）固定部分：这部分空间与所处理数据的大小和个数无关，主要包括程序代码、常量、简单变量、定长成分的结构变量所占的空间。

2）可变部分：这部分空间与算法在某次执行中处理的特定数据的大小和规模有关。

1.3 综合例题

【例 1-1】 设 n 为整数，求下面程序段的时间复杂度。

```
i=1; k=0;
 do
 { k+=10*i;    i++ ;
 } while (i<=n-1);
```

解：循环次数为 n-1，因此，语句执行的频度为 n-1，时间复杂度为 O(n)。

【例 1-2】 设 n 为整数，求下面程序段的时间复杂度。

```
x=0;
  for ( i=1: i<=n; i++)
  {     for ( j=1; j<=i; j++ )
            {   for ( k=1; k<=j; k++)
                      x+=delta;
            }
  }
```

解：这是一个三重循环，对固定的 i 及固定的 j，语句执行 j 次。

j 从 1 到 i 时语句执行次数：$\sum_{j=1}^{i} j=\frac{i(i+1)}{2}$

i 从 1 到 n 时语句执行次数：

$$\sum_{i=1}^{n}\frac{i(i+1)}{2}=\frac{1}{2}\left(\sum_{i=1}^{n}i^2+\sum_{i=1}^{n}i\right)=\frac{1}{2}\left(\frac{n(n+1)(2n+1)}{6}+\frac{n(n+1)}{2}\right)=\frac{n(n+1)(n+2)}{6}$$

因此语句执行的频度为：$\frac{n(n+1)(n+2)}{6}$，时间复杂度为 $O(n^3)$。

【例 1-3】 求如下两个 n 阶矩阵相乘运算程序段的时间复杂度。

```
for (i=1, i<=n; i++)
  {   for ( j=1; j<=n; j++)
```

```
    {   c[ i] [ j ]=0;                              /*1*/
        for ( k=1; k<=n ; k++)
          c [ i ] [ j ]=c [i ] [ j ] +a [i ][ k ] * b [ k ] [ j ];      /*2/
    }
}
```

解：因为/*1*/ 基本语句的语句频度为 n^2，/*2/基本语句的语句频度为 n^3。因此，程序段的时间复杂度为 $O(n^3)$。

1.4 小结

本章重点介绍了数据类型、数据结构等基本概念，简要讨论了几种基本数据结构的逻辑特征和几种常用存储结构，并介绍了算法的概念、算法的特性，时间性能和空间性能评价方法等。

1.5 习题

1．数据结构按逻辑结构可分为两大类，分别为_____。

2．线性结构中元素之间存在_____关系，树型结构中元素之间存_____ 关系，图型结构中元素之间存在_____ 关系。

3．一个算法应具有_____5 个特性。

4．一个算法的时间复杂度通常用它的_____形式表示，当一个算法的时间复杂度与问题的规模 n 大小无关时，则表示为_____ ；成正比时，则表示为_____；成对数关系时，则表示为_____。

5．一个算法的效率可为_____效率和_____效率。

6．数据结构被形式定义为(D, R)，其中 D 是（　　）的有限集合，R 是（　　）有限集合。

A．算法　　B．数据元素　　C．数据操作　　D．关系

7．数据在计算机存储器内表示时，物理地址与逻辑地址不相同的，称为（　　）。

A．存储结构　　B．逻辑结构　　C．链式存储结构　　D．顺序存储结构

8．算法分析的目的是（　　），算法分析的两个主要方面是（　　）。

A．给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．空间复杂度和时间复杂度

9．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．问题的规模和待处理数据的初态　　D．数据复杂度

10．计算机内部数据处理的基本单元（　　）。

A．数据　　B．数据元素　　C．数据项　　D．数据库

11．下面程序段的时间复杂度是多少？