



高等学校教材配套辅导
计算机类

数据结构

教材辅导

主编 王显

编写 计算机类教材辅导委员会

 科学技术文献出版社

高等学校教材配套辅导(计算机类)

数据结构

教材辅导

主 编
编 写
编写人员

王 显

计算机类教材辅导委员会

牟玉涛

顾 佳

李莉川

陈 楠

陈 丰

张怀甫

高永军

狄 懿

李 伟

李竞超

常慧敏

康建明

杨 军

张士新

苗红亮

王振凯

李 涛

张渭义

苏 燕

韩 彬

祝贺梅

谢检平

胡东华

杨 峰

蔡贵娟

李利娟

阎 伟

高 睿

李 芹

杜清平

韩 珍

周丽红

温 晴

张晓帆

钟崇光

高 鑫

温桂荣

韩 琴

郭海权

史 进

科学技术文献出版社

Scientific and Technical Documents Publishing House

· 北京 ·



图书在版编目(CIP)数据

数据结构教材辅导/王显主编. 北京:科学技术文献出版社,2008.11
ISBN 978-7-5023-3544-1

I. 数… II. 王… III. 数据结构-高等学校-教学参考资料 IV. TP311.
12

中国版本图书馆 CIP 数据核字(2008)第 153596 号

出 版 者	科学技术文献出版社
地 址	北京市复兴路 15 号(中央电视台西侧)/100038
图书编务部电话	(010)51501739
图书发行部电话	(010)51501720,(010)51501722(传真)
邮 购 部 电 话	(010)51501729
网 址	http://www.stdph.com
策 划 编 辑	科 文
责 任 编 辑	袁其兴 杜 娟
责 任 校 对	赵文珍
责 任 出 版	王杰馨
发 行 者	科学技术文献出版社发行 全国各地新华书店经销
印 刷 者	北京高迪印刷有限公司
版 (印) 次	2008 年 11 月修订版 第 1 次印刷
开 本	850×1168 32 开
字 数	225 千
印 张	7.75
印 数	1~5000 册
定 价	13.00 元

© 版权所有 违法必究

购买本社图书,凡字迹不清、缺页、倒页、脱页者,本社发行部负责调换。

声明:本书封面及封底均采用双博士品牌专用图标(见右图);该图标已由国家商标局注册登记。未经策划人同意,禁止其他单位或个人使用。



P 前言 Preface

“双博士”成就双博士!

本丛书的编写,以普通高等学校普遍采用的教材为蓝本,针对性强,信息含量高,具有很高的参考价值和实用意义,是考研专业课不可多得的工具与助手。

缺乏对专业课命题侧重点及考试要求的了解,已成为众多考生专业课考试失利的原因,进而与继续深造的机会失之交臂。因此,选取一本好的专业课辅导教材,对于有志于考研的莘莘学子来说,至关重要。本丛书涉及法学、金融、经管、通信电子、计算机、机械、控制理论与控制工程及其他热门专业。本书与市场上同类书相比,在内容编写方面更加细致详尽。在编排上分三部分:

1. **基本概念及考点精要:**对与本章相关的知识点进行串讲,使考生既能熟练掌握基础知识,又可把握重点、要点。

2. **典型例题、考题分析:**这一部分精选了各名校近年最新考研真题作为本书的例题,并提供详细的解析过程,强调解题思路。本部分内容既可使考生把握命题原则,又可熟悉题目类型,触类旁通。

3. **自测题及模拟训练题:**该部分为考生自行练习而提供,备有详细的解答过程。便于考生及时总结,查缺补漏。

本书附录为模拟试题,这些模拟试卷也是名校近几年的考试真题,并配有详细解析,具有非常典型的意义。

综合起来,本书凸显以下特色:

1. **专题化的编写体例:**面对普通高等学校专业课教材的泛泛的讲解,本书从更深的层次,对常考的知识点加重了讲解的力度,并与最新考试动态同步,及时补充了最新的考试内容。

2. 极富针对性的题型训练:在每章或每部分的典型例题、模拟试题中,均编排名校近几年的考研真题,并附有详细的参考答案,实战性极强。

3. 反映各名校最新考试信息:每章后所附的自测题及全书最后所附的全真模拟试卷,均选自各高校近几年考研真题,具有很高的参考价值。

策划本丛书的指导精神是既方便于在校本科生同步学习时参考,更适合于准备参加硕士研究生入学考试的学生作为专业课辅导用书使用。

温馨提示:

✱“双博士品牌图书”是全国最大的大学教辅图书和考研图书品牌,全国有三分之一的大学生和考研学生正在使用“双博士品牌图书”。

✱来自北京大学研究生会的感谢信摘要:双博士,您好!……,首先感谢您对北京大学的热情支持和无私帮助!双博士作为大学教学辅导和考研领域全国最大的图书品牌之一,不忘北大莘莘学子和传道授业的老师,其行为将永久被北大师生感怀和铭记! 北京大学研究生会

✱现在市场上有人冒用我们的书名,企图以假乱真,因此,读者在购买时,请认准双博士品牌。

编者

2008年于北京大学

目 录

第1章 绪论	(1)
1.1 基本概念及考点精要	(1)
1.2 典型例题、考题分析	(9)
1.3 自测题及模拟训练题	(9)
自测题参考答案	(10)
第2章 线性表	(11)
2.1 基本概念及考点精要	(11)
2.2 典型例题、考题分析	(42)
2.3 自测题及模拟训练题	(45)
自测题参考答案	(46)
第3章 栈和队列	(48)
3.1 基本概念及考点精要	(48)
3.2 典型例题、考题分析	(63)
3.3 自测题及模拟训练题	(66)
自测题参考答案	(67)
第4章 数组和广义表	(68)
4.1 基本概念及考点精要	(68)
4.2 典型例题、考题分析	(81)
4.3 自测题及模拟训练题	(83)
自测题参考答案	(84)
第5章 树与森林	(86)
5.1 基本概念及考点精要	(86)
5.2 典型例题、考题分析	(112)
5.3 自测题及模拟训练题	(114)
自测题参考答案	(116)
第6章 图	(119)
6.1 基本概念及考点精要	(119)

目 录

(1)	6.2 典型例题、考题分析	(145)
(1)	6.3 自测题及模拟训练题	(146)
(9)	自测题参考答案	(147)
(9)		
(01)	第7章 查 找	(148)
	7.1 基本概念及考点精要	(148)
	7.2 典型例题、考题分析	(176)
	7.3 自测题及模拟训练题	(178)
	自测题参考答案	(180)
	第8章 排 序	(183)
(81)	8.1 基本概念及考点精要	(183)
(81)	8.2 典型例题、考题分析	(211)
(89)	8.3 自测题及模拟训练题	(215)
(66)	自测题参考答案	(218)
(76)		
	第9章 硕士研究生入学考试全真模拟试卷及详解	
(80)		(220)
(88)	模拟试卷一	(220)
(88)	模拟试卷二	(228)
(18)	模拟试卷三	(235)
(83)		
(84)		
(88)		
(88)		
(311)		
(411)		
(811)		
(811)		
(811)		

第1章 绪论

核心考点:数据结构的定义;算法的特性和分析;算法的时间复杂度。

重点:数据结构中的几个重要概念和术语;算法设计的基本要求以及算法复杂度的分析和计算方法。

考试重点程度:★★

1.1 基本概念及考点精要

1.1.1 基本概念

1. 概念和术语

数据(Data):是对信息的一种符号表示。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。

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。一个数据元素可由若干个数据项组成。数据项是数据的不可分割的最小单位。

数据对象:是具有相同性质的数据元素的集合,是数据的一个子集。或某种数据类型元素的集合。数据对象可以是有限的,也可以是无限的。

例如整数的数据对象是 $\{\dots -3, -2, -1, 0, 1, 2, 3, \dots\}$

英文字符类型的数据对象是 $\{A, B, C, D, E, F, \dots\}$

数据结构不同于数据类型,也不同于数据对象,它不仅要描述数据类型和数据对象,而且要描述数据对象各元素之间的相互关系。

数据结构:按照一定逻辑关系组织,并按一定存储方法存储的数据的集合,且需要定义一系列运算。逻辑结构、存储结构和运算合称数据结构的三要素。

逻辑结构:规定了数据元素之间的逻辑关系,可用一个二元组表示:

$$\text{Data_Structure}=(D,R)$$

其中 D——数据元素的有限集合;

R——D上关系的有限集合。

例如 复数的数据结构定义如下:

$$\text{Complex}=(C,R)$$

其中,C 是含两个实数的集合 $\{C_1,C_2\}$,分别表示复数的实部和虚部。 $R=\{P\}$,P 是定义在集合上的一种关系 $\{<C_1,C_2>\}$ 。存储结构(或称为物理结构):指数据的逻辑结构在计算机存储器中的存储映像表示,即在能够反映数据元素逻辑关系的原则下,数据在存储器中的存储方式。

数据类型:在一种程序设计语言中,变量所具有的数据种类。

例如在 FORTRAN 语言中,变量的数据类型有整型、实型、和复数型。

例如在 C 语言中

数据类型:基本类型和构造类型

基本类型:整型、浮点型、字符型

构造类型:数组、结构、联合、指针、枚举型、自定义。

抽象数据类型(Abstract Data Type, ADT):是指基于一个逻辑类型的数据类型,以及这个数据类型上的一组操作。即只考虑逻辑结构和运算,而不考虑数据结构的存储结构。

抽象数据类型实际上就是对该数据结构的定义。因为它定义了一个数据的逻辑结构以及在此结构上的一组算法。用三元组描述如下:

$$\text{Data_structure}(D,R,P)$$

其中 P 为定义在该数据结构上的算法集合。

算法:建立在数据结构基础上的、求解问题的一系列确切的有限步骤。

2. 数据结构的研究目的和研究内容

数据结构的研究目的是有效地组织和处理非数值类型数据的理论、技术和方法,这类数据具有输入规模大,且具有一定内在逻辑关系的特点。

数据结构的核心研究内容包括3个方面:数据的逻辑结构、存储结构以及相应的基本操作运算的定义和实现。

3. 逻辑结构的4种基本形态

现实世界中非数值数据对象的逻辑关系可以划分为以下5种基本结构,如图1-1所示。它们的复杂程度依次增加(例如,以学校图书馆藏书作为数据对象,数据元素是图书的相关信息档案,考察数据元素之间的关系)。

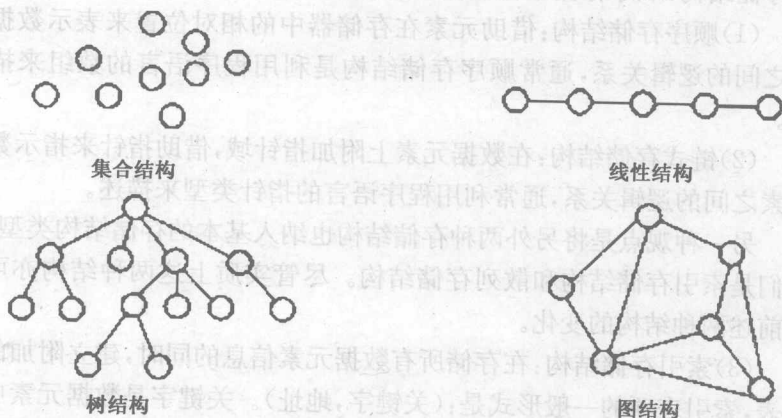


图 1-1 四类基本逻辑结构关系图

(1)集合结构:数据元素之间除了“属于同一集合”的联系之外,没有其他关系。例如,认定一本图书是否为库本。

(2)线性结构:数据元素之间存在一对一的关系。例如,以图书出版日期的时间先后顺序排列数据元素。

(3)树形结构:数据元素之间存在一个对多个的关系。例如,图书馆的馆藏图书分类目录体系,某类图书含多个子类,每个子类包含多本图书等……

(4)图状结构(或称网状结构):数据元素之间存在多对多的关系。例如,馆藏图书与阅览室的存放关系。

(5)文件结构:其存储方式本质上是线性结构,但其索引通常是树结构。

由于集合结构简单松散,且可以用其他结构来表示,因此教科书中的数据结构通常只讨论中间3种逻辑结构。其中树形结构和图状结构都属于非线性结构。

4. 数据存储结构的基本组织方式

数据结构在计算机中的表示称为数据的物理结构,又称为存储结构,有两种不同的表示方法:顺序表示和非顺序表示,由此得出两种不同的存储结构:顺序存储结构和链式存储结构。

(1)顺序存储结构: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,通常顺序存储结构是利用程序语言的数组来描述的。

(2)链式存储结构:在数据元素上附加指针域,借助指针来指示数据元素之间的逻辑关系,通常利用程序语言的指针类型来描述。

另一种观点是将另外两种存储结构也纳入基本的存储结构类型中,它们是索引存储结构和散列存储结构。尽管实质上这两种结构亦可看作前述两种结构的变化。

(3)索引存储结构:在存储所有数据元素信息的同时,建立附加的索引表,索引表项的一般形式是:(关键字,地址)。关键字是数据元素中可区分不同数据元素的数据项的值,通过关键字可以找到相关的数据元素的存储地址。

(4)散列存储结构(也称为 Hash 存储结构):此方法的基本思想是根据数据元素的关键字直接计算出相应的存储地址。

上述基本的存储结构既可以单独使用,也可以混合使用,选择何种应视具体要求而定,主要考虑的是操作运算方便以及算法的时间与空间要求。

5. 数据逻辑结构上定义的基本运算

基本运算的功能和数目以及每个基本运算中参数的数目和类型,都

应依据该数据结构的实际用途和需要来定义,定义是灵活的,不一定要照搬教材中的定义,理解这一点非常重要。基本的操作运算只有在一定的存储结构上具体实现之后才有真实的意义,这时使用者就可以按照抽象定义的形式来使用它们了,使用时和高级语言程序设计中的系统函数有相似之处,都不必关心该定义是如何实现的。基本的运算通常有以下几种:

- (1)建立数据结构:建立某种指定的数据结构。
- (2)清除数据结构:置某个指定的数据结构为空。
- (3)插入:在数据结构的指定位置上增加一个新的数据元素。
- (4)删除:在数据结构的指定位置上删除一个数据元素。
- (5)更新:修改数据结构中某个数据元素的值。
- (6)查找:在数据结构中寻找满足某种条件的数据元素。
- (7)排序:使数据元素按某种指定的次序重新排列(一般是在线性结构中)。
- (8)判空和判满:判定某个数据结构是否为空和判定该数据结构是否已达到逻辑上或存储上限制的最大允许容量。
- (9)求长:求指定的数据结构中的数据元素的个数。

6. 在数据结构中引入抽象数据类型概念的好处

抽象数据类型概念的运用实际上是将数据结构的讨论分成两部分:一部分是其概念所涵盖的数据逻辑结构的说明和抽象运算的定义,回答的是某种关系在数据对象上可以做什么;另一部分是在计算机中如何存储这些数据并如何具体实现抽象定义的运算,解决的是怎样做的问题。抽象数据类型面向使用层次,使使用层次对程序员透明化,主要的优点在于当具体实现方法改变时,不会影响用户的使用,可以提高含有该抽象数据类型的软件模块的复用度。抽象数据类型概念的思想与面向对象方法的思想是一致的。

7. 算法的5个重要特性

算法是对特定问题求解步骤的一种描述,是指令的有限序列,其中每一条指令表示一个或多个操作。算法具有以下5个特性:

- (1)有穷性:一个算法必须总是在执行有穷步之后结束,且每一步都

在有穷时间内完成。

(2) 确定性: 算法中每一条指令必须有确切的含义, 不存在二义性, 即算法只具有一个入口和一个出口。

(3) 可行性: 一个算法是可行的, 即算法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。

(4) 输入: 一个算法有零个或多个输入, 这些输入取自于某个特定的对象集合。

(5) 输出: 一个算法有一个或多个输出, 这些输出是同输入有着某些特定关系的量。

8. 算法设计的要求

评价一个好的算法有以下几个基本标准:

(1) 正确性(Correctness): 能够确保对于某种相对程度的随机输入有正确的输出, 即算法应满足具体问题的需求。

(2) 可读性(Readability): 算法描述清晰易懂, 便于修改和移植, 有利于阅读者对程序的理解。

(3) 健壮性(Robustness): 算法应具有容错处理。当输入非法数据时, 算法应对其作出适当的反应和处理, 而不是产生莫名其妙的输出结果。

(4) 效率(Effectiveness): 效率指的是算法执行的时间。算法设计合理, 执行时间效率高, 可以用时间复杂度度量。

(5) 存储量需求: 存储量需求指算法执行过程中所需要的最大存储空间。算法占用的存储容量应该合理, 可以用时间复杂度度量。一般地, 效率和存储量需求与问题的规模有关。

9. 算法分析的目的

算法的时间复杂度和空间复杂度分析构成了算法分析的目的是考察算法的时间和空间效率, 以求改进算法或对不同的算法效率进行比较。内存运算空间一般情况下较为充足, 我们把算法的时间复杂度分析作为重点的讨论对象。

10. 算法时间复杂度的含义

语句的频度是指该语句在算法中被重复执行的次数。算法中所有

语句的频率之和记做 $T(n)$, 它是该算法所求解问题规模 n 的函数。当问题的规模趋向无穷大时, $T(n)$ 的数量级(阶)称为渐近时间复杂度, 简称为时间复杂度, 记作 $T(n)=O(f(n))$ 。

上述表达式中“ O ”的文字含义是 $T(n)$ 的量级, 其严格的数学定义是: 若 $T(n)$ 和 $f(n)$ 是定义在正整数集合上的两个函数, 则存在正的常数 C 和 n_0 , 使得当 $n \geq n_0$ 时, 都满足 $0 \leq T(n) \leq C \cdot f(n)$ 。

例如: $\{++x; s=0;\}$

将 X 自增看成是基本操作, 则语句频率为 1, 即时间复杂度为 $O(1)$ 。

如果将 $S=0$ 也看成是基本操作, 则语句频率为 2, 其时间复杂度仍为 $O(1)$, 即常量阶。

例如: $\text{for}(i=1; i \leq n; ++i) \{ ++x; S+=x; \}$

语句频率为 $2n$, 其时间复杂度为 $O(n)$, 即时间复杂度为线性阶。

例如: $\text{for}(i=1; i \leq n; ++i)$

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

$\{$

$\quad c[i][j]=0;$

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

$\quad \quad c[i][j] += a[i][k] * b[k][j];$

$\}$

由于是一个三重循环, 每个循环从 1 到 n , 则总次数为: $n \times n \times n = n^3$, 时间复杂度为 $T(n)=O(n^3)$

定理: 若 $A(n) = a_m n^m + a_{m-1} n^{m-1} + \dots + a_1 n + a_0$ 是一个 m 次多项式, 则 $A(n)=O(n^m)$

证略。

例如: $\text{for}(i=2; i \leq n; ++i)$

$\quad \text{for}(j=2; j \leq i-1; ++j)$

$\quad \quad \{ ++x; a[i][j]=x; \}$

语句频率为:

$$1+2+3+\dots+n-2 = (1+n-2) \times (n-2) / 2$$

当 n 趋向无穷大时 $= (n-1)(n-2)/2$ 趋向无穷大
 $= n^2 - 3n + 2$

∴ 时间复杂度为 $O(n^2)$

即此算法的时间复杂度为平方阶。

常有一个算法时间复杂度为 $O(1)$ 的算法, 它的基本运算执行的次数是固定的。因此, 总的时间由一个常数(即零次多项式)来限界。而一个时间为 $O(n^2)$ 的算法则由一个二次多项式来限界。

以下 6 种计算算法时间复杂度的多项式是最常用的。其关系为:

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

指数时间的关系为:

$$O(2^n) < O(n!) < O(n^n)$$

当 n 取得很大时, 指数时间算法和多项式时间算法在所需时间上非常悬殊。因此, 只要有人能将现有指数时间算法中的任何一个算法化简为多项式时间算法, 那就取得了一个伟大的成就。

有的情况下, 算法中基本操作重复执行的次数还随问题的输入数据集不同而不同。

例如:

```
void bubble-sort(int a[], int n) {
    for (i = n - 1; i > 1; i--) {
        change = false;
        for (j = 0; j < i; j++) {
            if (a[j] > a[j + 1]) {
                swap(a[j], a[j + 1]); // 交换 a[j] 和 a[j + 1]
                change = true;
            }
        }
    }
}
```

最好情况: 0 次

最坏情况: $1 + 2 + 3 + \dots + n - 1 = n(n - 1) / 2$

平均时间复杂度为: $O(n^2)$

11. 算法空间复杂度的含义

空间复杂度是对一个算法在运行过程中产生的临时数据占用的存储空间(或称辅助空间)大小的量度,一般也是所求解问题规模 n 的函数,以数量级形式给出,记作 $S(n)=O(f(n))$ 。

若辅助空间是常数,则空间复杂度为 $O(1)$ 。

在计算空间复杂度不便的情况下,有时也以存储密度来研究算法的空间使用状况。存储密度 d 定义为:

$$d = \frac{\text{数据本身所占内存容量}}{\text{数据实际存储所占内存容量}}$$

1.2 典型例题、考题分析

【例题 1-1】 (西安电子科技大学计算机应用专业 1998 年试题)
数据结构是研究数据的_____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_____,而确保经过这些运算后所得到的新结构是_____结构类型。

【解析】

本部分基本上都是概念试题。

【答案】

物理结构、逻辑结构、运算、算法、原来的

【考点解析】

本章涉及的基本概念较多,历来是研究生入学考试概念题的命题点。

1.3 自测题及模拟训练题

【习题 1-1】 (北京师范大学 2008 年)简述算法的特性。

自测题参考答案

【习题 1-1】

【解析】

此题为基础题,考查对算法的理解。

【答案】

- (1)有穷性 (2)确定性 (3)可行性 (4)输入 (5)输出