

计算机核心课程辅导系列教材

数据结构

题解 · 综合练习

赵宇 邱镭 编著



 机械工业出版社
CHINA MACHINE PRESS

计算机核心课程辅导系列教材

数据结构题解·综合练习

赵宇 邱镭 编著



机械工业出版社

本辅导教材是针对“计算机应用、计算机信息管理”专业课程，并结合数据结构教学大纲编写而成的。

全书共 11 章，前 10 章中每章先介绍该章的主要知识点和重点内容，对基本概念和方法作了扼要的讲解；然后给出习题与参考答案，其中，习题包括选择题、填空题和思考题三部分。第 11 章为综合练习。

本书可作为《数据结构》教材的辅导用书，也可以作为大专院校相关专业学生的参考书。

图书在版编目 (CIP) 数据

数据结构题解·综合练习/赵宇, 邱镭编著. —北京: 机械工业出版社, 2004.7
(计算机核心课程辅导系列教材)

ISBN 7-111-14905-X

I. 数... II. ①赵...②邱... III. 数据结构—习题 IV. TP311.12-44

中国版本图书馆 CIP 数据核字 (2004) 第 069511 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 李利健

责任印制: 李 妍

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2004 年 9 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·9.75 印张·240 千字

0001—5000 册

定价: 16.00 元

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

出版说明

高等教育是培养人才过程中非常重要的一个环节。近年来随着高等教育的不断发展，作为计算机、信息管理等相关专业，每年接受各种教育和培训的人数不下百万，包括在读大专院校的学生、参加高等教育自学考试的学员、参加全国计算机等级考试的人员以及参加研究生入学考试的考生。国家也有许多这方面的考试，如软件人员水平测试、企业信息管理师考试等。

作为计算机、信息管理等专业学生必修的一些专业基础课和专业课程，在学习过程中会遇到很多问题和困难。为了帮助学习者学好和掌握课程内容，正确和深入地理解和把握课程内容，并在参加相关课程的各种考试时能够考出好成绩，我们组织了一批具有相关课程教学经验的一线教师，编写了这套“计算机核心课程辅导系列教材”。

本系列教材中所包括的课程都属于学科重点课程、核心课程和考研课程。每本书都是针对相关课程编写的，不仅仅是与某一具体教材配套的辅导书。本系列教材采用全新的体例结构。内容包括：重点难点分析、练习题及综合练习，并附有参考答案，有些课程还配有实验。编写体例也突破了一般指导书的模式，在对概念、原理内容的描述上尽量采用要点分解方式，使学习者一目了然，直接深入到概念和原理的本质和核心中，便于记忆、掌握和灵活应用，对提高应试成绩具有很大的帮助。由于本系列教材已对课程内容进行了充分的整理和提炼，所以本系列教材还是学生学习过程中最好的“笔记”。考虑到学生在学习中有记笔记的需要，在主要内容、重点内容部分的版面设计中，为读者留出了记笔记的地方，进一步方便了读者的使用。

本系列教材注重概念、原理的准确性，精练性，以及内容的全面性。高校在校内可根据所使用的教材，将其作为辅导材料。对考研的读者，这套辅导教材也是非常好的参考书。另外，本系列教材还适合自考学生和参加各类信息技术考试的人员作参考。

前 言

《数据结构》是计算机应用、计算机信息管理专业的一门重要的专业基础课程，主要讲述数据结构和算法。

本书是《数据结构》教材的辅导教材，全书共 11 章，分为两个部分。第一部分由第 1~10 章组成，每章根据教学大纲的要求，按节对主要知识点和重点内容进行了归纳和总结。第 1 章主要讲解数据结构的概念和类型等问题；第 2 章主要讲解关于线性表的逻辑结构和存储及运算等问题；第 3 章主要讲解栈和队列；第 4 章主要讲解串的基本运算、存储等问题；第 5 章主要讲解多维数组和广义表；第 6 章主要讲解二叉树的定义、性质、操作、存储等问题；第 7 章主要讲解图；第 8 章主要讲解各种排序；第 9 章主要讲解有关查找的各种方法；第 10 章主要讲解有关文件的问题。第二部分即第 11 章提供了三套综合练习，供学生练习。

本书概念清晰、通俗明了，适合作为《数据结构》教材的辅导用书，也可作为大专院校相关专业学生的参考书。

本书由赵宇（编写第 1~7 章）、邱镭（编写第 8~11 章）编写。参与部分编写工作的有高颖瑄、赵维珩等。另外，本书在编写过程中还得到了盛定宇教授、彭澎副教授、马亚铃高级工程师的帮助。

由于编者水平有限，书中不妥之处在所难免，恳请读者批评指正。

编 者

目 录

出版说明	
前言	

第1章 概论	1
第1节 基本概念和术语	2
第2节 学习数据结构的意义	3
第3节 算法的描述和分析	3
练习题	4
参考答案	6
第2章 线性表	8
第1节 线性表的逻辑结构	9
第2节 线性表的顺序存储结构	10
第3节 线性表的链式存储结构	12
第4节 顺序表和链式表的比较	18
练习题	19
参考答案	22
第3章 栈和队列	24
第1节 栈	25
第2节 队列	28
练习题	31
参考答案	34
第4章 串	36
第1节 串及其运算	37
第2节 串的存储结构	37
练习题	40
参考答案	43
第5章 多维数组和广义表	46
第1节 多维数组	47
第2节 矩阵的压缩存储	48
第3节 广义表	50

练习题	52
参考答案	54
第6章 树	58
第1节 树的概念	59
第2节 二叉树	60
第3节 二叉树的遍历	62
第4节 线索二叉树	64
第5节 树和森林	65
第6节 哈夫曼树及其应用	67
练习题	70
参考答案	74
第7章 图	77
第1节 图的相关概念	78
第2节 图的存储结构	79
第3节 图的遍历	81
第4节 生成树和最小生成树	82
第5节 最短路径	84
第6节 拓扑排序	86
练习题	88
参考答案	91
第8章 排序	95
第1节 排序概念	96
第2节 插入排序	97
第3节 交换排序	98
第4节 选择排序	100
第5节 归并排序	103
第6节 分配排序	103
第7节 各种内部排序方法的比较和选择	104
练习题	105
参考答案	108
第9章 查找	111
第1节 基本概念	112

第2节 线性表的查找	112	第6节 多关键字文件	129
第3节 树上的查找	115	练习题	130
第4节 散列技术	117	参考答案	132
练习题	120		
参考答案	123		
第10章 文件	126	第11章 综合练习	133
第1节 文件的基本概念	127	综合练习(一)	134
第2节 顺序文件	128	综合练习(二)	137
第3节 索引文件	128	综合练习(三)	140
第4节 索引顺序文件	128	综合练习题(一) 参考答案	143
第5节 散列文件	129	综合练习题(二) 参考答案	145
		综合练习题(三) 参考答案	147

第1章

概论

- 基本概念和术语
- 学习数据结构的意义
- 算法的描述和分析

本章介绍了数据、数据元素、数据结构（包括逻辑结构、存储结构）以及数据类型的概念、数据的逻辑结构分类，及其逻辑特征、数据的存储结构可用的几种基本存储方法。时间复杂度与渐近时间复杂度的概念，如何求算法的时间复杂度等。

通过本章的学习，要求掌握数据、数据元素、数据项、数据结构等基本概念，数据结构的逻辑结构、存储结构及数据运算的含义及其关系，数据结构的两类逻辑结构和四种常用的存储表示方法；数据结构在各种软件系统中所起的作用，选择合适的数据结构是解决应用问题的关键步骤；算法、算法的时间复杂度和空间复杂度、最坏的和平均的时间复杂度等概念，算法的时间复杂度不仅仅依赖于问题的规模，也取决于输入实例的初始化状态，算法描述和算法分析的方法，对于一般算法能分析出时间复杂度。需要特别注意的是算法复杂度的分析方法。

第1节 基本概念和术语

笔记

1. 数据的定义

数据（Data）是信息的载体，它能够被计算机识别、存储和加工处理。

2. 数据结构的定义

数据结构（Data Structure）是指相互之间存在着一种或多种关系的数据元素的集合。包含：

- 1) 数据之间的逻辑关系；
- 2) 数据元素及其在计算机存储器内的表示；
- 3) 数据的运算。

3. 数据类型的定义

1) 数据类型（Data Type）：是一个值的集合和定义在这个值集上的一组操作的总称。

2) 抽象数据类型：抽象数据的组织和与之相关的操作。

4. 数据逻辑结构分类

数据的逻辑结构分为线性结构（线性表、栈、队列）、非线性结构（多维数组、树、图）。

5. 数据存储结构分类

数据的存储结构分为：顺序存储结构、链式存储结构、索引存储结构、散列存储结构。

1) 顺序存储结构：指把逻辑上相邻的结点存储在物理位置上相邻的存储单元里，结点间的逻辑关系由存储单元的临界关系来体现，这种存储表示称作顺序存储结构。

2) 链式存储结构：指不要求逻辑上相邻的结点在物理位置上也相

邻，结点间的逻辑关系是有附加的指针字段表示的，这种存储表示称作链式存储结构。

3) 索引存储结构：指除建立存储结点信息外，还建立附加的索引表来标识结点的地址。组成索引表的索引项由结点的关键字和地址组成。包括稠密索引和稀疏索引。

① 稠密索引 (Dense Index)：指若每个结点在索引表中都有一个索引项，则该索引表称为稠密索引。

② 稀疏索引：指若一组结点在索引表中只对应一个索引项，则该索引表称为稀疏索引。

4) 散列存储结构：指根据结点的关键字直接计算出该结点的存储地址。

6. 表

一个表（数据库）称为一个数据结构，由很多记录（数据元素）组成，每个元素又包括很多字段（数据项）组成。对于表中的任一个记录（结点），它只有一个直接前趋和一个直接后继（前趋后继就是前相邻后相邻的意思），整个表只有一个开始结点和一个终端结点。

第2节 学习数据结构的意义

1. 数据结构的核心理念

数据结构的核心理念是分解与抽象。通过分解可以划分出数据的三个层次；再通过抽象，舍弃数据元素的具体内容，得到逻辑结构。

这是一个从具体（即具体问题）到抽象（即数据结构）的过程。

熟练地掌握这两个过程是数据结构课程在专业技能培养方面的基本目标。

2. 学习数据结构的意义

1) 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。

2) 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促进学生的综合应用能力和专业素质的提高。

笔 记

第3节 算法的描述和分析

1. 算法的特性

算法 (Algorithm) 是任意一个良定义的计算过程，它以一个或多个值作为输入，并产生一个或多个值作为输出。因此，算法是一系列将输入转化成输出的计算步骤，一个算法也可以被认为是用来解决一个计算问题的工具，计算问题的陈述指定了所想要的输入输出之间的关系，

笔 记

笔 记

而相应的算法则描述了达到这种关系的计算过程。

一个算法必须满足五个重要的特性：有穷性、确定性、可行性、有输入、有输出。

要设计一个好的算法通常要考虑以下的要求：

- 1) 正确。算法的执行结果应当满足预先规定的功能和性能要求。
- 2) 可读。一个算法应当思路清晰、层次分明、简单明了、易读易懂。
- 3) 健壮。输入不合法数据时，应能作适当处理，不至引起严重后果。
- 4) 高效。有效使用存储空间和有较高的时间效率。

选择一个算法，要考虑以下几个方面：

- 1) 算法首先应当是正确的。
- 2) 执行算法所需要耗费的时间。
- 3) 执行算法所耗费的存储空间（主要是辅助存储空间）。
- 4) 算法应便于理解、编码、调试等。

2. 算法的时间复杂度

算法的时间复杂度（Time complexity）是指程序运行从开始到结束所需要的时间。

算法的平均时间复杂度指所有可能的输入实例都以等概率的情况出现时，算法的预期运行时间。

算法的最坏时间复杂度指算法在任何输入实例上运行时间的上界，也就是说要确保算法的运行时间不会比它更长。

通常用 $O(1)$ 表示常数计算时间。常见的渐进时间复杂度有：

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

3. 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时所需的存储量。

程序的一次运行是针对所求解问题的某一特定实例而言的。例如，求解排序问题中每次执行排序算法是对一组特定个数的元素进行排序，对该组元素的排序是排序问题的一个实例，元素个数可视为该实例的特征。

程序运行所需要的存储空间包括以下两部分：

1) 固定部分：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；空间与所处理数据的大小和个数无关，或者称与问题实例的特征无关。

2) 可变部分：分空间大小与算法在某次执行中处理的特定数据的大小和规模有关。

练习题

一、单项选择题 在下列每个题的 4 个备选答案中选出 1 个正确答案，将答案的正确号码填在题中的括号内。

参考答案

题目及备选答案

- C** 1. 计算机算法必须具备输入、输出和 ()。
A. 计算方法 B. 排序方法
C. 解决问题的有限运算步骤 D. 程序设计方法
- B** 2. 若对数据结构采用了顺序存储, 第一个节点的地址为 1001, 每个节点的值需占用 2 个存储单元, 则第三个节点的起始地址为 ()。
A. 1003 B. 1005
C. 1006 D. 1007
- C** 3. 设三个函数 f, g, h 分别为 $f(n)=100n^3+n^2+1000$, $g(n) = 25n^3 + 5000n^2$, $h(n)=n^{1.5}+5000n\lg n$, 则下列关系不成立的是 ()。
A. $h(n)=O(n^{1.5})$ B. $g(n)=O(f(n))$
C. $h(n)=O(n\lg n)$ D. $f(n)=O(g(n))$
- B** 4. () 是数据的基本单位。
A. 数据结构 B. 数据元素
C. 数据项 D. 数据类型
- C** 5. 一般而言, 最适合描述算法的语言是 ()。
A. 自然语言 B. 计算机程序语言
C. 伪语言 D. 数学公式
- C** 6. 某程序的时间复杂度为 $(14n+n\log_2 n+n^2+37)$, 其数量级表示为 ()。
A. $O(n)$ B. $O(n\log_2 n)$
C. $O(n^2)$ D. $O(\log_2 n)$

二、填空题 将正确答案填写到横线处, 不填或填错均不得分。

参考答案

题 目

- 逻辑结构** 1. 数据结构的实质一般包括三个方面的内容: 数据的_____, 数据的存储结构及_____。
- 存储结构数据的运算** 2. 数据的_____结构是独立于计算机的。
- 逻辑** 3. 一个算法的空间复杂度是指该算法所耗费的_____, 它是该算法求解问题_____的函数。
- 存储空间** 4. 数据元素是数据的_____。
- 规模 n** 5. 数据的逻辑结构是对数据之间_____的描述。
- 基本单位** 6. 数据的_____结构是依赖于计算机的。
- 逻辑关系** 7. 一般, 将算法求解问题的输入量称为_____。
- 存储** 8. 线性表中第一个节点没有直接前趋, 称为_____。
- 问题的规模** 9. 一个算法应具备的 5 个特性为_____, _____、_____, _____、_____。
- 开始节点**
- 有穷性、确定性、可行性、输入、输出**

顺序结构、索引 10. 数据的存储结构被分为_____、链式结构、_____和散列结构四种。

三、名词解释

1. 表
2. 算法
3. 算法的时间复杂度
4. 算法的空间复杂度
5. 数据结构

四、思考题

1. 按增长率由小至大的顺序排列下列各函数: 1^{160} , $\left(\frac{2}{3}\right)^n$, $\left(\frac{3}{2}\right)^n$, n^n , $n!$, 2^n , $\lg n$, $n^{\frac{3}{2}}$
2. 常用的存储表示方法有哪几种?
3. 要设计一个好的算法通常要考虑哪些要求?

参考答案

三、名词解释

1. 表

一个表(数据库)称为一个数据结构,由很多记录(数据元素)组成,每个元素又包括很多字段(数据项)组成。对于表中的任一记录(结点),它只有一个直接前趋和一个直接后继(前趋后继就是前相邻后相邻的意思),整个表也只有一个开始结点和一个终端结点。

2. 算法

算法(Algorithm)是任意一个良定义的计算过程,它以一个或多个值作为输入,并产生一个或多个值作为输出。因此算法是一系列将输入转化成输出的计算步骤,一个算法也可以被认为用来解决一个计算问题的工具,计算问题的陈述指定了所渴望的输入输出之间的关系,而相应的算法则描述了达到这种关系的计算过程。

3. 算法的时间复杂度

算法的时间复杂度(Time complexity)指程序运行从开始到结束所需要的时间。

4. 算法的空间复杂度

算法的空间复杂度(Space complexity)指程序运行从开始到结束所需的存储量。

5. 数据结构

数据结构(Data Structure)是指相互之间存在着一种或多种关系的数据元素的集合。包含数据之间的逻辑关系;数据元素及其在计算机存储器内的表示;数据的运算。

四、思考题

1. 按增长率由小至大的顺序排列下列各函数: 1^{160} , $\left(\frac{2}{3}\right)^n$, $\left(\frac{3}{2}\right)^n$, n^n , $n!$, 2^n , $\lg n$, $n^{\frac{3}{2}}$

答: 由小至大的顺序排列: $\left(\frac{2}{3}\right)^n < 1^{160} < \lg n < n^{\frac{3}{2}} < \left(\frac{3}{2}\right)^n < 2^n < n! < n^n$

2. 常用的存储表示方法有哪几种?

答: 常用的存储表示方法有四种:

1) 顺序存储方法: 它是把逻辑上相邻的结点存储在物理位置相邻的存储单元里, 结点间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储表示称为顺序存储结构, 通常借助程序语言的数组描述。

2) 链式存储方法: 它不要求逻辑上相邻的结点在物理位置上亦相邻, 结点间的逻辑关系是由附加的指针字段表示。由此得到的存储表示称为链式存储结构, 通常借助于程序语言的指针类型描述。

3) 索引存储方法: 除建立存储结点信息外, 还建立附加的索引表来标识结点的地址。组成索引表的索引项由结点的关键字和地址组成。若每个结点在索引表中都有一个索引项, 则该索引表称之为稠密索引 (Dense Index); 若一组结点在索引表中只对应一个索引项, 则该索引表称为稀疏索引。

4) 散列存储方法: 就是根据结点的关键字直接计算出该结点的存储地址。

3. 要设计一个好的算法通常要考虑哪些要求?

答: 要设计一个好的算法通常要考虑以下的要求:

1) 正确。算法的执行结果应当满足预先规定的功能和性能要求;

2) 可读。一个算法应当思路清晰、层次分明、简单明了、易读易懂;

3) 健壮。当输入不合法数据时, 应能作适当处理, 不至引起严重后果。

4) 高效。有效使用存储空间和有较高的时间效率。

2

第2章

线性表

- 线性表的逻辑结构
- 线性表的顺序存储结构
- 线性表的链式存储结构
- 顺序表和链表的比较

本章介绍了线性表的逻辑结构特征、常见的线性表的几种基本运算；顺序表的特征、顺序表中结点地址的计算；顺序表上实现的基本运算；单链表的特征、图形表示法；单链表的各种算法实现；循环链表的特征；双链表的特征以及它们的主要算法实现等。

通过本章的学习，要求掌握线性表的逻辑结构特征，线性表上定义的基本运算，并能利用基本运算构造出复杂的运算；顺序表的含义及特点，即顺序表如何反映线性表中元素之间的逻辑关系；顺序表上的插入、删除操作及其平均时间性能分析；利用顺序表设计算法解决的应用问题；链表如何表示线性表中元素之间的逻辑关系；链表中头指针和头结点的作用；单链表、双链表、循环链表链接方式上的区别；单链表上实现的建表、查找、插入和删除等基本算法，并分析其时间复杂度；循环链表上尾指针取代头指针的作用，以及单循环链表上的算法与单链表上相应算法的异同点；双向链表的定义及其相关算法；利用链表设计算法解决简单的应用问题；顺序表和链表的主要优缺点；针对线性表上所需要执行的主要操作，知道选择顺序表还是链表作为其存储结构才能达到较优的时空性能。在本章学习过程中，需要特别注意涉及有效算法解决与线性表相关的应用问题。

第1节 线性表的逻辑结构

1. 线性表的定义

线性表是具有相同数据类型的 $n(n \geq 0)$ 个数据元素的有限序列，通常记为： $(a_1, a_2, \dots, a_n)$ ，其中， n 为表长， $n=0$ 时称为空表。

2. 线性表的特点

一种逻辑结构的数据元素属于相同数据类型，它们之间的关系是线性关系。

3. 常见线性表的基本运算

线性表上的基本操作有以下 6 类。

(1) 线性表初始化：Init_List(L)

初始条件：表 L 不存在。

操作结果：构造一个空的线性表。

(2) 求线性表的长度：Length_List(L)

初始条件：表 L 存在。

操作结果：返回线性表中的所含元素的个数。

(3) 取表元：Get_List(L,i)

初始条件：表 L 存在且 $1 \leq i \leq \text{Length_List}(L)$ 。

操作结果：返回线性表 L 中的第 i 个元素的值或地址。

(4) 按值查找：Locate_List(L,x)，x 是给定的一个数据元素

初始条件：线性表 L 存在。

操作结果：在表 L 中查找值为 x 的数据元素，其结果返回在 L 中

笔记

首次出现的值为 x 的那个元素的序号或地址，称为查找成功；否则，在 L 中未找到值为 x 的数据元素，返回一特殊值表示查找失败。

(5) 插入操作: $\text{Insert_List}(L, i, x)$

初始条件: 线性表 L 存在，插入位置正确 ($1 \leq i \leq n+1$, n 为插入前的表长)。

操作结果: 在线性表 L 的第 i 个位置上插入一个值为 x 的新元素，这样使原序号为 $i, i+1, \dots, n$ 的数据元素的序号变为 $i+1, i+2, \dots, n+1$ ，插入后表长=原表长+1。

(6) 删除操作: $\text{Delete_List}(L, i)$

初始条件: 线性表 L 存在， $1 \leq i \leq n$ 。

操作结果: 在线性表 L 中删除序号为 i 的数据元素，删除后使序号为 $i+1, i+2, \dots, n$ 的元素变为序号为 $i, i+1, \dots, n-1$ ，新表长=原表长-1。

第2节 线性表的顺序存储结构

笔记

1. 顺序表

(1) 顺序表的定义

顺序存储的线性表按线性表中元素的逻辑顺序，依次存放在地址连续的存储单元里，其存储特点是用物理上的相邻实现逻辑上的相邻。

(2) 顺序表的特点

1) 顺序表是由地址连续的向量实现的，具有按序号随机访问的特点。

2) 存储密度高。

3) 做插入和删除运算时，需移动表中大约一半的元素。

4) 顺序表的存储空间是静态分配的，在程序执行之前必须明确规定它的存储规模，因此，分配不足则造成溢出；分配过大，可能造成存储单元的浪费。

(3) 具体实现方法

在程序设计语言中，用在内存中占用连续存储区域的一维数组来表示顺序表。考虑到顺序表表长是可变的（插入、删除等运算），因此，数组的容量设计需足够大，例如，用 $\text{data}[\text{ListSize}]$ 来表示顺序表，其中 ListSize 是一个根据实际问题定义的足够大的整数，表中的数据从 $\text{data}[0]$ 开始依次顺序存放，但当前线性表中的实际元素个数可能未达到最大的 ListSize 个，因此需用一个变量 length 来记录当前线性表中实际元素的个数，当表空时， $\text{length}=0$ 。

方法一：用一个数组和表示长度的变量共同完成上述思想，例：

```
DataType data[ListSize];
```

```
int length
```

数据元素分别存放在 $\text{data}[0]$ 到 $\text{data}[\text{length}-1]$ 中（C 语言默认方式，