

基础教育系列



21世纪高校计算机应用技术系列规划教材

丛书主编 谭浩强

实用数据结构基础学习指导

陈元春 王淮亭 编著

44



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

基础教育系列

21 世纪高校计算机应用技术系列规划教材
丛书主编 谭浩强



实用数据结构基础学习指导

陈元春 王淮亭 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是《实用数据结构基础（第二版）》（陈元春、张亮、王勇编著，中国铁道出版社出版）的配套学习指导书，全书内容分为四个部分：教学内容指导（包括对主教材第1章~第10章的知识点分析、典型习题分析和各章单元练习解答）、自主设计实验指导（对10个自主设计实验的设计思想、算法分析进行了详细的指导）、模拟试卷（6套）和模拟试卷参考答案。

本书对数据结构的概念和原理的阐述通俗易懂，例子翔实；习题难易适中，题型丰富；对数据结构基本运算的分析注重实现的过程。本书以C/C++语言作为算法的描述语言，对于书中的实验和一些重要的算法均给出了完整的C/C++语言源程序，并全部在Visual C++ 6.0环境下运行通过。

本书适合作为计算机应用专业本科和高职高专数据结构学科的教辅书，同时也可以作为成人教育、自学考试和从事计算机应用的工程技术人员的参考书。

图书在版编目（CIP）数据

实用数据结构基础学习指导 / 陈元春，王淮亭编著.

北京：中国铁道出版社，2008.10

（21世纪高校计算机应用技术系列规划教材·基础教育系列）

ISBN 978-7-113-09219-1

I. 实… II. ①陈…②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152582号

书 名：实用数据结构基础学习指导

作 者：陈元春 王淮亭 编著

策划编辑：严晓舟 秦绪好

责任编辑：崔晓静

编辑部电话：(010) 63583215

编辑助理：孟欣

封面制作：白雪

责任印制：李佳

出版发行：中国铁道出版社（北京市宣武区右安门西街8号 邮政编码：100054）

印 刷：中国铁道出版社印刷厂

版 次：2008年11月第1版 2008年11月第1次印刷

开 本：787mm×1092mm 1/16 印张：13.5 字数：310千

印 数：5 000册

书 号：ISBN 978-7-113-09219-1/TP·2990

定 价：22.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签，无标签者不得销售

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

21 世纪高校计算机应用技术系列规划教材

主任：谭浩强

副主任：陈维兴 严晓舟

委员：（按姓氏音序排列）

安淑芝	安志远	陈志泊	韩 劼	侯冬梅
李 宁	李雁翎	林成春	刘宇君	秦建中
秦绪好	曲建民	尚晓航	邵丽萍	宋 红
宋金珂	王兴玲	魏善沛	熊伟建	薛淑斌
张 玲	赵乃真	訾秀玲		

序

PREFACE

21 世纪是信息技术高度发展且得到广泛应用的时代,信息技术从多方面改变着人类的生活、工作和思维方式。每一个人都应当学习信息技术、应用信息技术。人们平常所说的计算机教育其内涵实际上已经发展为信息技术教育,内容主要包括计算机和网络的基本知识及应用。

对多数人来说,学习计算机的目的是为了利用这个现代化工具工作或处理面临的各种问题,使自己能够跟上时代前进的步伐,同时在学习的过程中努力培养自己的信息素养,使自己具有信息时代所要求的科学素质,站在信息技术发展和应用的前列,推动我国信息技术的发展。

学习计算机课程有两种不同的方法:一是从理论入手;二是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说,不仅需要“知道什么”,更重要的是“会做什么”。因此,在学习过程中要以应用为目的,注重培养应用能力,大力加强实践环节,激励创新意识。

根据实际教学的需要,我们组织编写了这套“21 世纪高校计算机应用技术系列规划教材”。顾名思义,这套教材的特点是突出应用技术,面向实际应用。在选材上,根据实际应用的需要决定内容的取舍,坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上,采取“提出问题—解决问题—归纳分析”的三部曲,这种从实际到理论、从具体到抽象、从个别到一般的方法,符合人们的认知规律,且在实践过程中已取得了很好的效果。

本套教材采取模块化的结构,根据需要确定一批书目,提供了一个课程菜单供各校选用,以后可根据信息技术的发展和教学的需要,不断地补充和调整。我们的指导思想是面向实际、面向应用、面向对象。只有这样,才能比较灵活地满足不同学校、不同专业的需要。在此,希望各校的老师们把你们的要求反映给我们,我们将会尽最大努力满足大家的要求。

本套教材可以作为大学计算机应用技术课程的教材以及高职高专、成人高校和面向社会的培训班的教材,也可作为学习计算机的自学教材。

由于全国各地、各高等院校的情况不同,因此需要有不同特点的教材以满足不同学校、不同专业教学的需要,尤其是高职高专教育发展迅速,不能照搬普通高校的教材和教学方法,必须要针对它们的特点组织教材和教学。因此,我们在原有基础上,对这套教材作了进一步的规划。

本套教材包括以下五个系列:

- 基础教育系列
- 高职高专系列
- 实训教程系列
- 案例汇编系列
- 试题汇编系列

其中，基础教育系列是面向应用型高校的教材，对象是普通高校的应用型专业的本科学生。高职高专系列是面向两年制或三年制的高职高专院校的学生的，突出实用技术和应用技能，不涉及过多的理论和概念，强调实践环节，学以致用。后面三个系列是辅助性的教材和参考书，可供应用型本科和高职学生选用。

本套教材自 2003 年出版以来，已出版了 70 多种，受到了许多高校师生的欢迎，其中有多种教材被国家教育部评为普通高等教育“十一五”国家级规划教材。《计算机应用基础》一书出版三年内发行了 50 万册。这表示了读者和社会对本系列教材的充分肯定，对我们是有力的鞭策。

本套教材由浩强创作室与中国铁道出版社共同策划，选择有丰富教学经验的普通高校老师和高职高专院校的老师编写。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版的过程中，得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持，对此谨表衷心的感谢。

本套教材如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本套教材的不断完善和出版，为我国计算机教育事业的发展和人才培养做出更大贡献。

全国高等院校计算机基础教育研究会会长
“21 世纪高校计算机应用技术系列规划教材”丛书主编

谭浩强

前言

FOREWORD

数据结构是计算机专业及相关专业的一门重要的专业基础课程。它不仅是计算机程序设计的理论基础，而且是学习计算机操作系统原理、编译原理、数据库原理等课程的重要基础。

数据结构的主要任务是讨论数据的各种逻辑结构和数据在计算机中的存储表示以及各种非数值运算算法的实现。通过数据结构课程的学习，学生能使用数据结构的基本分析方法来提高编写程序的能力和用计算机解决实际问题的能力。

由于数据结构学科所涉及的概念、原理和方法比较抽象，再加上知识点多、难度大，对于初学者来说不容易理解和掌握，尤其对于 C 语言基础较差和程序设计能力较弱的学生更是如此。不少学生在解答数据结构习题时，往往感到无从着手，更不知道算法如何描述。作者根据多年的教学经验积累，并收集、整理了大量文献，编写了本书，目的是通过知识点复习、典型习题分析，使学生充分掌握数据结构的原理，开拓求解数据结构问题的思路，提高分析问题和解决问题的能力，为编写数据结构的算法打下坚实的基础。

本书是《实用数据结构基础（第二版）》（陈元春、张亮、王勇编著，中国铁道出版社出版）的配套学习指导书，全书内容分为四个部分。

第一部分为教学内容指导，包括对主教材第 1 章～第 10 章的知识点分析、典型习题分析和各章单元练习解答。

第二部分为自主设计实验指导，对 10 个自主设计实验的设计思想、算法分析进行了详细的指导，并给出了完整的 C 程序源代码。所有源程序都在 Visual C++ 6.0 环境下运行通过。通过这些实验，可以使学生了解并学会如何运用数据结构的知识去解决实际问题，并培养设计较复杂算法的基本能力。

第三部分为模拟试卷，包含了 6 套模拟试卷。模拟试卷主要是为了帮助学生在学完数据结构课程以后自我检验。每套试题中有判断题、填空题、选择题、应用题、程序填空题、算法分析题和程序设计题等题型。试题覆盖了教材中的大部分知识点，其目的是帮助学生对数据结构课程进行系统复习和自我测试。

第四部分为模拟试卷参考答案，供学生参考。

本书第一部分和第二部分的第 1 章、第 3 章、第 4 章、第 6 章、第 7 章以及第三部分、第四部分由陈元春编写；第一部分和第二部分的第 2 章、第 5 章、第 8 章～第 10 章由王淮亭编写；陈默绘制了本书的大量插图，在此一并表示感谢！

本书适合作为使用《实用数据结构基础（第二版）》教材的学生的参考资料，也可以作为数据结构自学者的参考书。

由于作者水平有限，加之成书时间仓促，书中的疏漏或不当之处在所难免，恳请广大专家和读者不吝赐教。

编者
2008年9月

目录

CONTENTS

第一部分 教学内容指导

第 1 章 绪论	1
1.1 知识点分析	1
1.2 典型习题分析	2
1.3 单元练习 1 解答	5
第 2 章 线性表	7
2.1 知识点分析	7
2.2 典型习题分析	8
2.3 单元练习 2 解答	14
第 3 章 栈	17
3.1 知识点分析	17
3.2 典型习题分析	19
3.3 单元练习 3 解答	22
第 4 章 队列	26
4.1 知识点分析	26
4.2 典型习题分析	27
4.3 单元练习 4 解答	31
第 5 章 串	35
5.1 知识点分析	35
5.2 典型习题分析	36
5.3 单元练习 5 解答	40
第 6 章 多维数组和广义表	45
6.1 知识点分析	45
6.2 典型习题分析	46
6.3 单元练习 6 解答	51
第 7 章 树和二叉树	54
7.1 知识点分析	54
7.2 典型习题分析	56
7.3 单元练习 7 解答	61
第 8 章 图	70
8.1 知识点分析	70
8.2 典型习题分析	71
8.3 单元练习 8 解答	76

第 9 章 查找	81
9.1 知识点分析	81
9.2 典型习题分析	82
9.3 单元练习 9 解答	89
第 10 章 排序	95
10.1 知识点分析	95
10.2 典型习题分析	97
10.3 单元练习 10 解答	101

第二部分 自主设计实验指导

实验 1 学生成绩分析程序	107
实验 2 多项式求和	112
实验 3 后缀表达式求值	118
实验 4 循环队列的实现和运算	125
实验 5 字符串分割处理	129
实验 6 稀疏矩阵十字链表的存储	133
实验 7 标识符树与表达式求值	140
实验 8 最小生成树	144
实验 9 哈希查找	147
实验 10 双向冒泡排序	152

第三部分 模拟试卷

模拟试卷 1	154
模拟试卷 2	158
模拟试卷 3	163
模拟试卷 4	167
模拟试卷 5	171
模拟试卷 6	177

第四部分 模拟试卷参考答案

模拟试卷 1 参考答案	184
模拟试卷 2 参考答案	188
模拟试卷 3 参考答案	191
模拟试卷 4 参考答案	194
模拟试卷 5 参考答案	197
模拟试卷 6 参考答案	200
参考文献	204

第一部分 教学内容指导

第 1 章 绪 论

1.1 知识点分析

1. 数据

数据是信息的载体，是对客观事物的符号表示。数据是计算机化的现实世界的事物的抽象描述。凡是能被计算机识别、存取和加工处理的符号、字符、图形、图像、声音、视频信号等一切信息都可以称为数据。

2. 数据元素

数据元素是对现实世界中某独立个体的数据描述，是数据的基本单位。数据元素也称为结点。

3. 数据项

数据项是数据不可分割的、具有独立意义的最小数据单位，是对数据元素属性的描述。

数据、数据元素、数据项反映了数据组织的三个层次，即数据可以由若干个数据元素组成，数据元素又由若干数据项组成。

4. 数据对象

数据对象是性质相同的数据元素的集合，是数据的一个子集。

5. 数据结构

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。简而言之，数据结构是指数据之间的关系，即数据的组织形式。数据结构包括以下三个方面：

(1) 数据的逻辑结构

数据元素之间的逻辑关系，称为数据的逻辑结构。即从逻辑关系上描述数据，它与数据的存储无关，是独立于计算机的。一个数据的逻辑结构 G 可以用二元组 $G=(D,R)$ 来表示，其中 D 是数据元素的集合， R 是 D 上所有数据元素之间关系的有限集合。

线性结构是指数据元素之间存在“一对一”关系的逻辑结构，非线性结构是指数据元素之间存在“一对多”或“多对多”关系的逻辑结构。

(2) 数据的存储结构

数据元素及其关系在计算机存储器内的表示，称为数据的存储结构，也称为数据的物理结构。数据的存储结构是数据的逻辑结构用计算机语言的实现，依赖于计算机语言。

(3) 数据的运算

数据的运算是指对数据施加的操作。数据的运算是定义在数据的逻辑结构上的，而运算的实现则是在存储结构上进行的。

6. 算法的描述和分析

数据的运算是通过算法来描述的，对于算法的说明可以使用不同的语言，对同一问题可以有不同的算法。首先选用的算法必须是正确的。其次，主要考虑执行算法所耗费的时间（即时间效率）和执行算法所耗费的存储空间（即空间效率）。

(1) 时间复杂度

通常，把算法中所包含简单操作次数的多少称为算法的时间复杂度。但是当—个算法比较复杂时，其时间复杂度的计算会变得相当困难。—般情况下，算法中原操作重复执行的次数是规模 n 的某个函数 $f(n)$ ，算法的时间复杂度 $T(n)$ 的数量级可记做 $T(n)=O(f(n))$ 。

算法的时间复杂度 $T(n)$ 是该算法的时间消耗，—个算法的时间耗费就是该算法中所有语句的执行次数（频度）之和。当 $n \rightarrow \infty$ 时（即当 n 相当大时）， $T(n)$ 的数量级（阶），用“ O ”表示。由于 $\lim T(n)/f(n)=C$ ， C 是不为 0 的常数，所以 $T(n)=O(f(n))$ 。其实， $f(n)$ 就是 $T(n)$ 中最高阶的一项，是算法中频度最大的语句频度。

—般情况下，对于循环语句只需要考虑循环体中语句的执行次数，而忽略该语句中循环头的部分。有时，循环体中语句的频度不仅与问题规模 n 有关，还与输入实例等其他因素有关，此时可以用最坏情况下的时间复杂度作为算法的时间复杂度。

(2) 空间复杂度

—个程序的空间复杂度是指程序运行从开始到结束所需要的存储空间。类似于算法的时间复杂度，我们把算法所需存储空间的量度，记做 $S(n)=O(f(n))$ 。其中， n 为问题的规模。—个程序上机执行时，除了需要存储空间来存放本身所用的指令、常数、变量和输入数据以外，还需要一些对数据进行操作的工作单元和实现算法所必需的辅助空间。在进行时间复杂度分析时，如果所占空间量依赖于特定的输入，一般都按最坏情况分析。

程序运行时所需要的存储空间包括以下两个部分：

① 固定部分：主要包括程序代码、常量、变量、结构体变量等所占的空间。空间与所处理的数据大小和个数无关，或者说与问题事例的特征无关。

② 可变部分：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。

1.2 典型习题分析

【例 1.1】 算法与程序的区别。

解： 算法与程序既有区别，又有联系。其区别是：

① 算法必须满足有穷性，而程序则不一定满足有穷性。例如，启动计算机必须使用操作系

统，只要不关机或不受破坏，操作系统就永不终止。即使没有作业运行，它也一直处在一个等待循环之中。因此，操作系统是一个不终止的计算过程，但它不满足算法的定义。

② 程序中的指令必须用计算机可以接受的语言书写，而算法则无此限制。但是，当用一个计算机可以接受的语言来书写算法时，它就是程序。一般而言，算法比较抽象，而程序则比较具体。

【例 1.2】通常一个算法的时间复杂度是指 ()。

- A. 算法的平均时间复杂度 B. 算法在最坏情况下的时间复杂度
C. 算法的期望运行时间 D. 算法在最好情况下的时间复杂度

分析：如果没有“平均”、“最好”、“最坏”的修饰语，时间复杂度就是指最坏的时间复杂度。最坏时间复杂度是算法的所有输入可能情况的执行时间的上界，所以应选 B。

解：B。

【例 1.3】当 n 取 1~10 时，比较 n 、 $2n$ 、 n^2 、 n^3 、 2^n 、 $n!$ 、 n^n 增长率的变化是什么？

解：如表 1-1-1 所示为当 n 取 1~10 时各函数的变化过程，可见当 $n \geq 10$ 时，按增长率由小到大排列依次为 n 、 $2n$ 、 n^2 、 n^3 、 2^n 、 $n!$ 、 n^n 。

表 1-1-1 各函数值增长表

n	$2n$	n^2	n^3	2^n	$n!$	n^n
1	2	1	1	2	1	1
2	4	4	8	4	2	4
3	6	9	27	8	6	27
4	8	16	64	16	24	256
5	10	25	125	32	120	3 125
6	12	36	216	64	720	46 656
7	14	49	343	128	5 040	823 543
8	16	64	512	256	40 320	16 777 216
9	18	81	792	512	362 880	3.9×10^8
10	20	100	1 000	1 024	3 628 800	1.0×10^{10}

【例 1.4】 $T(n)=n^{\sin n}$ ，则用“O”可以表示为 ()。

- A. $T(n)=O(n^{-1})$ B. $T(n)=O(1)$ C. $T(n)=O(n)$ D. 不确定

分析： $\sin n$ 的取值范围是 $-1 \sim 1$ ，所以 $T(n)$ 的上界为 $O(n^1)$ ，即 $O(n)$ ，所以应选 C。

解：C。

【例 1.5】设两个算法的执行时间分别为 $100n^2$ 和 2^n ，它们在同一台计算机上运行，要使前者快于后者， n 至少要多大？

分析：要使前者快于后者，即求满足 $100n^2 < 2^n$ 的 n 。由于 $100n^2$ 和 2^n 这两个函数都是单调递增函数，随着 n 的增大， 2^n 的递增速度比 $100n^2$ 快。在 $n \geq 1$ 的整数情况下，可得出当 $n \geq 15$ 时， $100n^2 < 2^n$ （可以编写一个程序进行测试）。所以，要使前者快于后者， n 至少要大于等于 15。

解： $n \geq 15$ 。

【例 1.6】当 n 充分大时, 按从小到大的次序对下列时间进行排序:

① $T_1(n)=5n^2+10n+6\lg n$ 。

② $T_2(n)=3n^2+100n+3\lg n$ 。

③ $T_3(n)=8n^2+3\lg n$ 。

④ $T_4(n)=2n^2+6\ 000n\lg n$ 。

分析: 为了比较两个同数量级算法的优劣, 需突出主项的常数因子, 而将低次的项用“ O ”表示, 则 $T_1(n)=5n^2+10n+6\lg n=5n^2+O(n)$, $T_2(n)=3n^2+100n+3\lg n=3n^2+O(n)$, $T_3(n)=8n^2+3\lg n=8n^2+O(\lg n)$, $T_4(n)=2n^2+6\ 000n\lg n=2n^2+O(n\lg n)$ 。当 n 足够大时, $T_1(n)$ 、 $T_2(n)$ 、 $T_3(n)$ 、 $T_4(n)$ 的时间复杂度都为 $O(n^2)$, 虽然它们的数量级相同, 但各自主项的系数还是有区别的。因为 $T_4(n)$ 的主项常数因子 $< T_2(n)$ 的主项常数因子 $< T_1(n)$ 的主项常数因子 $< T_3(n)$ 的主项常数因子, 所以从小到大的顺序为 $T_4(n)$ 、 $T_2(n)$ 、 $T_1(n)$ 、 $T_3(n)$ 。

解: $T_4(n)$ 、 $T_2(n)$ 、 $T_1(n)$ 、 $T_3(n)$ 。

【例 1.7】设三个函数 $f(x)$ 、 $g(x)$ 、 $h(x)$ 分别为: $f(n)=100n^3+n^2+100$ 、 $g(n)=25n^3+50n^2$ 、 $h(n)=n^{1.5}+50n\lg n$, 判断下列关系是否成立:

① $f(n)=O(g(n))$ 。 ② $h(n)=O(n^{1.5})$ 。 ③ $h(n)=O(n\lg n)$ 。

分析: ① $\lim f(n)/g(n)=\lim(100n^3+n^2+100)/(25n^3+50n^2)=4$, 即 $f(n)$ 和 $g(n)$ 的数量级相同, 所以① $f(n)=O(g(n))$ 成立。

② $h(n)$ 的数量级取决于 $n^{1.5}$ 和 $n\lg n$, 50 是与 n 无关的常数因子, 可以忽略。因为 $n^{1.5}$ 的数量级大于 $\lg n$, 也大于 $n\lg n$, 即 $\lim h(n)/n^{1.5}=\lim(n^{1.5}+50n\lg n)/n^{1.5}=1$, 所以② $h(n)=O(n^{1.5})$ 成立。

③ $\lim h(n)/n\lg n=\lim(n^{1.5}+50n\lg n)/n\lg n=\infty$, 所以③ $h(n)=O(n\lg n)$ 不成立。

解: ① 成立。② 成立。③ 不成立。

【例 1.8】将一维数组中的元素逆置的算法如下, 试分析其时间频度及时间复杂度。

【程序代码】

```
void exchange(int a[],int n)
{
    for(int i=0;i<=n/2-1;i++)
    {
        int t=a[i];
        a[i]=a[n-i-1];
        a[n-i-1]=t;
    }
}
```

解: 时间频度为 $T(n)=3n/2$, 当 n 充分大时, n 的系数 $3/2$ 可以忽略不计, 所以其时间复杂度为 $O(n)$ 。

【例 1.9】将 n 个元素按升序排列的算法如下, 试分析其时间频度及时间复杂度。

【程序代码】

```
void sort(int a[],int n)
{
    int i,j,k,t;
    for(i=0;i<n-1;j++)
    {
        k=i;
        for(j=i+1;j<=n-1;j++)
```

```

    if(a[k]>a[j])
        k=j;
    if(k!=i)
    {
        t=a[k];
        a[k]=a[i];
        a[i]=t;
    }
}
}

```

解：时间复杂度为 $T(n)=1+(n-1)+(1+2+3+\dots+n-1)+3(n-1)=n^2/2+7n/2-3$ ，时间复杂度为 $O(n^2)$ 。

【例 1.10】设 n 为正整数，分析下列程序段中加下画线的语句的程序步数。

【程序代码】

```

int i=1;
do{
    for(int j=1;j<=n;j++)
        i=i+j;
}while(i<100+n)

```

分析： $i=1$ 结束时， $i=1+n(n+1)/2$ ； $i=2$ 结束时， $i=(1+n(n+1)/2)+n(n+1)/2=1+2(n(n+1)/2)$ ； $i=3$ 结束时， $i=(1+2(n(n+1)/2))+n(n+1)/2=1+3(n(n+1)/2)$ 。一般来说， $i=k$ 结束时， $i=1+k(n(n+1)/2)<100+n$ ，求出满足此不等式的 k 的最大值，语句 $i=i+j$ 的程序步数为： $(k+1) \times (n(n+1)/2)$ 。

解： $(k+1) \times (n(n+1)/2)$ 。

1.3 单元练习 1 解答

一、判断题答案

题目	1	2	3	4	5	6	7	8	9	10
答案	√	√	×	×	×	√	√	√	×	√

二、填空题答案

- | | | |
|------------------|-----------|----------------------|
| (1) 存储结构 | (2) 图形结构 | (3) 非线性结构 |
| (4) 树形结构 图形结构 | (5) 1 | (6) 任意多个 |
| (7) 数据的物理结构 | (8) 散列存储 | (9) 一对一 |
| (10) 一对多 | (11) 多对多 | (12) 算法 (或运算) |
| (13) 所有数据元素之间的关系 | (14) 有穷指令 | (15) 事后统计法 |
| (16) 输入规模 | (17) 存储空间 | (18) $O(n \log_2 n)$ |
| (19) $O(n^2)$ | (20) 操作对象 | |

三、选择题答案

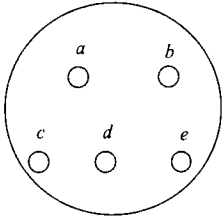
题目	1	2	3	4	5	6	7	8	9	10
答案	A	C	C	D	A	C	B	A	B	D
题目	11	12	13	14	15	16	17	18	19	20
答案	C	A	A	C	A	C	D	D	A	C

四、答案

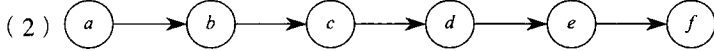
(1) $O(n \times m)$ (2) $O(n^2)$ (3) $O(1)$ (4) $O(n)$ (5) $O(n^2)$

五、答案

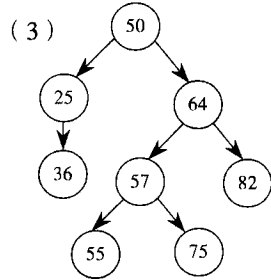
(1)



属于集合。

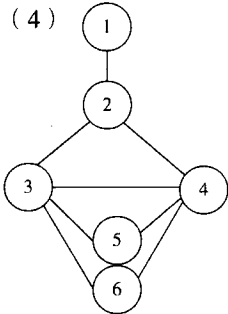


属于线性结构。



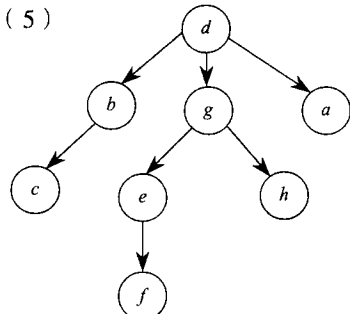
属于树结构。

(4)



属于图结构。

(5)



属于树结构。

第 2 章 线性表

2.1 知识点分析

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:

$$(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$$

其中, n 为表长, $n=0$ 时称为空表。

在线性表中相邻元素之间存在着顺序关系。对于元素 a_i 而言, a_{i-1} 称为 a_i 的直接前驱, a_{i+1} 称为 a_i 的直接后继。也就是说:

- ① 有且仅有一个开始结点 (a_1), 它没有直接前驱。
- ② 有且仅有一个终端结点 (a_n), 它没有直接后继。
- ③ 除了开始结点和终端结点以外, 其余的结点都有且仅有一个直接前驱和一个直接后继。

2. 顺序表

顺序表是线性表的顺序存储, 是指在内存中分配一组连续地址的存储单元依次存储线性表的数据元素, 元素之间的结构 (关系) 无须存放, 顺序表的逻辑顺序和物理顺序是一致的。

常用的顺序表的操作 (运算) 有建立顺序表、插入、删除、查找等。

3. 线性链表

线性链表 (单链表) 是线性表的链接式存储, 在内存中分配的存储单元可以是连续的地址, 也可以是不连续的地址, 存储单元中存放数据元素信息 (数据域) 和存放其后继结点地址 (指针域)。

常用的线性链表操作 (运算) 有建立链表、求表长、查找、插入、删除等。

4. 顺序表和线性链表的比较

(1) 顺序表

优点: 空间利用率高; 可以随机存取表中任意一个元素, 存储位置可以用公式 $B+(i-1) \times d$ 计算。其中, B 为基地址。

缺点: 对顺序表做插入、删除时需要通过移动大量的数据元素, 时间性能差; 线性表预先分配空间时, 必须按最大空间分配, 存储空间得不到充分的利用; 表的容量难以扩充 (对有些高级语言而言)。

(2) 单链表

优点：对表做插入、删除时无须移动数据元素，时间性能好；因为动态分配存储空间，不会发生溢出现象。

缺点：存储密度小，空间利用率低；查找元素必须从首元素开始查找，比较费时。

2.2 典型习题分析

【例 2.1】带头结点的单链表 L 为空的判断条件是 ()。

- A. $L == \text{NULL}$ B. $L \rightarrow \text{next} == \text{NULL}$ C. $L \rightarrow \text{next} == L$ D. $L != \text{NULL}$

分析：带头结点的单链表为空表如图 1-2-1 所示。所以，其判断条件可以表示为 $L \rightarrow \text{next} == \text{NULL}$ (注意用“==”)，如果单链表不带头结点，则其空判断条件是 $L == \text{NULL}$ ，如果是循环单链表，其空判断条件是 $L \rightarrow \text{next} == L$ ，如图 1-2-2 所示，故答案为 B。

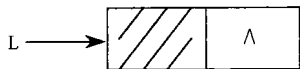


图 1-2-1 带头结点的单链表

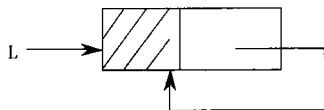


图 1-2-2 带头结点的循环单链表

解：B。

【例 2.2】在 n 个结点的顺序表中，算法的时间复杂度是 $O(1)$ 的操作是 ()。

- A. 访问第 i 个结点 ($1 \leq i \leq n$) 和求第 i 个结点的直接前驱 ($2 \leq i \leq n$)
 B. 在第 i 个结点之后插入一个新结点 ($1 \leq i \leq n$)
 C. 删除第 i 个结点 ($1 \leq i \leq n$)
 D. 将 n 个结点从小到大排序

分析：在顺序表中第 i 个结点之后插入一个新结点 ($1 \leq i \leq n$)，在其后结点都要顺序往后移动，所以时间复杂度为 $O(n)$ ，B 不能选。顺序表删除第 i 个结点 ($1 \leq i \leq n$)，在其后结点都要顺序往前移动，所以时间复杂度为 $O(n)$ ，C 不能选。在顺序表中，将 n 个结点从小到大排序，时间复杂度大于 $O(n)$ ，D 不能选。在顺序表中访问第 i 个结点，其地址可有公式直接计算出，因此时间复杂度为 $O(1)$ ，同样求第 i 结点的直接前驱第 $i-1$ 结点，其时间复杂度也为 $O(1)$ ，故选 A。

解：A。

【例 2.3】在具有 n 个结点的单链表中，其算法的时间复杂度是 $O(n)$ 的操作是 ()。

- A. 求链表的第 i 个结点 B. 在地址为 p 的结点之后插入一个结点
 C. 删除开始结点 D. 删除地址为 p 的结点的后继结点

分析：由单链表知在地址为 p 的结点之后插入一个结点，如图 1-2-3 所示，只需修改指针，即① $s \rightarrow \text{next} = p \rightarrow \text{next}$ ；② $p \rightarrow \text{next} = s$ ；时间复杂度是 $O(1)$ ，B 不能选。删除开始结点只需要修改头结点， $H \rightarrow \text{next} = H \rightarrow \text{next} \rightarrow \text{next}$ ；就可以了，时间复杂度是 $O(1)$ ，C 不能选。删除地址为 p 的结点的后继结点，如图 1-2-4 所示， $p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next}$ ；就可以，时间复杂度是 $O(1)$ ，D 不能选。求链表的第 i 个结点，必须从头结点开始，如图 1-2-5 所示，其时间复杂度是 $O(n)$ ，故选 A。