



计算机专业
考研辅导丛书

数据结构

联考辅导教程

(2013版)

李春葆 尹为民 蒋晶珏 编著

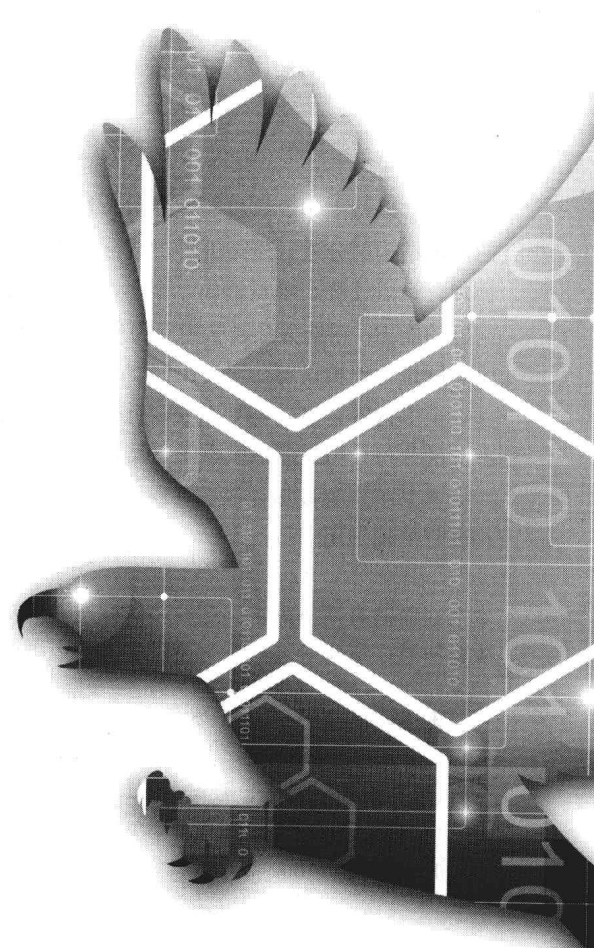


- **名师指点**：学科知识点归纳，难点解析，重点复习
- **阅卷专家点评**：历年经典试卷详解，近年真题剖析，2012年真题评析
- **模拟试卷练习**：考前热身、自测与冲刺

通过备考复习为将来的**创新**打下**扎实**的理论基础



清华大学出版社



计算机专业
考研辅导丛书

数据结构

联考辅导教程

+++++
+++++
+++++

(2013版)

李春葆 尹为民 蒋晶珏 编著

清华大学出版社
北京

内 容 简 介

编者以几十年计算机专业学科教育积累的丰富经验,针对近几版全国计算机学科专业基础综合考试大纲的数据结构部分进行透彻剖析,结合编者参加4年全国联考阅卷工作的体会,对数据结构的内容进行知识点梳理、疑点诠释、难点辅导、全面复习;通过详解大量例题的各种求解方法,力求帮助提高考生分析与解决问题的能力;精选大量教学中广为采用的用例、历年名校考研试题以及近4年的考研真题和最新的2012年考研真题进行评析讲解,对考生存在的一些问题,在写作上力求具有指导性和针对性。

全书内容丰富,辅导专业,针对性强,所有考纲中的知识点都标识了难度和重要性,另给出了数据结构部分的5份模拟试题以及参考答案,供考生练习、参阅。

本书可作为参加计算机专业研究生入学考试的考生复习用书,也可以作为计算机专业的学生学习数据结构课程的辅导用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构联考辅导教程:2013版/李春葆,尹为民,蒋晶珏编著. —北京:清华大学出版社,2012.6

(计算机专业考研辅导丛书)

ISBN 978-7-302-28780-3

I. ①数… II. ①李… ②尹… ③蒋… III. ①数据结构—研究生—入学考试—自学参考资料
IV. ①TP311.12

中国版本图书馆CIP数据核字(2012)第089429号

责任编辑:夏非彼

封面设计:王 翔

责任校对:闫秀华

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:190mm×260mm 印 张:22.25 字 数:570千字

版 次:2012年6月第1版 印 次:2012年6月第1次印刷

印 数:1~6000

定 价:42.00元

产品编号:047627-01

全国计算机专业研究生入学统一考试于 2009 年开始实施, 随后《全国硕士研究生入学统一考试计算机科学与技术学科联考计算机学科专业基础综合考试大纲》2009 年至 2012 年 4 个版本相继发布, 其对考试科目、考试范围、考试要求、考试形式和试卷结构进行了系统规定。

针对 2013 年的计算机专业联考, 我们组织专家和一线教师对 4 个版本考纲进行了深入剖析, 并在对自身教学经验进行认真总结、提炼之后, 推出了这套“计算机专业考研辅导丛书”, 丛书共 5 本:

- 计算机学科专业基础综合联考辅导教程 (2013 版)
- 数据结构联考辅导教程 (2013 版)
- 计算机组成原理联考辅导教程 (2013 版)
- 计算机操作系统联考辅导教程 (2013 版)
- 计算机网络联考辅导教程 (2013 版)

本丛书具有如下特色。

精准剖析考纲内容

深入研讨考研命题的方向和动态, 结合近 4 年出题情况, 总结出计算机学科各专业的复习要求。丛书以考纲知识点为中心, 汇总了各知识点近 4 年全国考研题的出题情况, 对每个知识点进行了深入地归纳和整理, 具有知识面广和综合性强等特点。

把握知识间的内在联系, 拓展创新思维

把握知识点之间的关系, 将掌握的知识变“活”。通过对知识点分解, 找出贯穿于各知识之间的内在联系, 并配上相关的例题, 阐明如何利用这些内在联系解决问题, 从而做到不仅授人以“鱼”, 更注重授人以“渔”。

全面复习, 形成知识体系

本丛书以各门考研课程的权威教材为依托, 对教材进行了全面、深入地剖析和提炼。在考研复习中, 往往一个微小的知识漏洞, 就可能造成无法弥补的损失, 因此复习必须扎实全面。丛书以基

本知识点归纳和例题分析为框架，构建了一个完备的知识体系，为考生全程领跑。

以典型例题分析带动能力培养

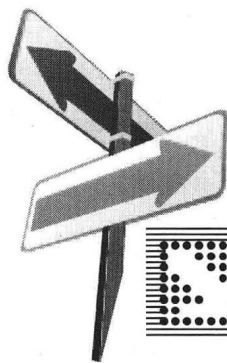
本丛书以典型题目的分析为突破口，注重点拨解题思路，强化各知识点的灵活运用，从而启发解题灵感。在单项选择题部分主要涵盖各门课程的核心知识点，在综合应用题部分重点强化各门课程知识点的交叉。不仅所有例题给出了参考答案，大部分还给出了详细透彻的分析过程，便于读者在解题过程中举一反三，触类旁通，进而提高分析问题和解决问题的能力。

最新考研真题解析与模拟试题练习

本丛书收集近年具有代表性的最新考研真题（包括 2012 年考研真题）进行分析，深化点评，精解考点；丛书中各本均提供 5 份模拟试题练习，并给出参考答案，供考生自测与考前冲刺训练。

本套丛书由长期坚持在教学第一线的教师编写，其中多位教师参加了近 4 年的全国考研阅卷工作，在编写中体现了他（她）们的教学经验和评卷见解，希望能够有助于考生提高考研成绩，培养本学科的综合分析能力；备考是目的，更重要的是通过复习为未来的创新打下扎实的理论基础。

本套丛书的编写工作得到武汉大学计算机学院王丽娜副院长、许先斌、郑鹏和余利等教授和老师的大力支持，教学办丁红丽老师也提供了帮助，清华大学出版社夏非彼等老师提供有指导性的建议，作者在此一并表示衷心感谢。



前言

数据结构课程是计算机专业的核心课程，主要讨论数据组织和数据处理方法。数据结构的考点较多，算法设计理论性较强，掌握起来比较困难。作为计算机专业基础综合试题的一部分，在总分 150 的试卷中占 45 分，在全国联考后，对考试范围做了进一步规范，不再包含外排序和文件内容。

迄今为止，全国计算机专业联考大纲发行了 4 版，就数据结构部分而言，考点的变化不大，2011 年考纲增加了 B+树部分，2012 年考纲增加了外部排序部分。

联考大纲中指定数据结构的考查目标是：掌握数据结构的基本概念、基本原理和基本方法；掌握数据的逻辑结构、存储结构及基本操作的实现，能够对算法进行基本的时间复杂度与空间复杂度的分析；能够运用数据结构的基本原理和方法进行问题的分析和求解，具备采用 C、C++或 Java 语言设计和实现算法的能力。就考查目标而言，联考大纲对运用基本原理分析解决问题的能力要求越来越高，这就要求考生透彻理解数据结构的基本知识和原理，注重算法设计与实现过程。

近 4 年的出题情况是：2009 年单项选择题共 10 题（20 分），综合应用题 2 题（10+15=25 分）；2010~2012 年均为单项选择题 11 题（22 分），综合应用题 2 题（10+13=23 分）。4 年的题型几乎相同。

编者结合联考大纲和数据结构课程的特点，并总结这 4 年参加全国联阅卷卷中考生出现的各种问题编写了本书。全书按联考大纲数据结构知识点的顺序进行组织安排，附录 A 给出了 5 份数据结构部分模拟试题，附录 B 给出了 5 份模拟试题的参考答案。

在编写过程中主要参考了《数据结构（C 语言版）》（严蔚敏、吴伟民编著，清华大学出版社，1997）、《数据结构教程（第 3 版）》（李春葆等编著，清华大学出版社，2009）和《数据结构习题与解析（B 版）》（李春葆等编著，清华大学出版社，2007）等，也参阅了多本较有影响的数据结构教材和教学辅导书，在此向这些作者们表示衷心感谢。

本书的特点如下：

- 对联考大纲进行深入剖析。不仅挖掘考纲中各知识点的内涵和解题技巧，还以★~★★★★★的方式给出了每个知识点的难易程度和重要性。
- 收集大量的例题并予以解析。不仅有数据结构在教学过程中广为采用的用例，还包括往

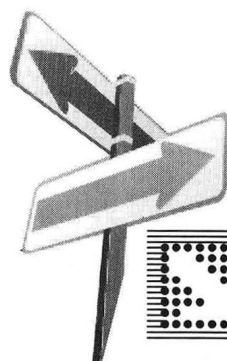
年部分著名高校的考研试题，也将 2009~2012 年 4 年的考研题按知识点分布归纳其中。并按★~★★★★★等级给出每道题的难易程度。根据全国考研题的题型，所有例题仅分为单项选择题和综合应用题。

- 结合近几年联考命题情况，给出了 5 份模拟试题并予以详解。

除本书列出的作者外，参加编写的人员还有安杨、喻丹丹、余云霞、董尚燕、伍春香、喻卫、杨波、苏亮、黄明辉、陈钉、程洁等老师。

在编写过程中，编者结合多年数据结构课程的教学经验和考研辅导中考生提出的问题，力求能够全面、透彻地解析各个考点。但由于水平所限，书中若有不当和错误之处，敬请读者指正。

编 者
2012 年 5 月



目 录

第 1 章 绪论	1
1.1 数据结构的基本概念	2
1.2 算法及其分析	7
1.3 递归算法设计	12
第 2 章 线性表	16
2.1 线性表的定义	17
2.2 顺序表	18
2.3 单链表	27
2.4 双链表	51
2.5 循环链表	57
2.6 有序表	64
第 3 章 栈、队列和数组	77
3.1 栈	78
3.2 队列	91
3.3 数组和稀疏矩阵	101
第 4 章 树与二叉树	106
4.1 树的概念	108
4.2 二叉树的概念	114
4.3 二叉树的遍历	123
4.4 二叉树的构造	164
4.5 树和二叉树的相互转换	170
4.6 线索二叉树	174

4.7 二叉排序树	178
4.8 平衡二叉树	188
4.9 哈夫曼树	198
第 5 章 图	201
5.1 图的基本概念	203
5.2 图的存储结构	207
5.3 图的遍历	216
5.4 最小生成树	237
5.5 最短路径	241
5.6 拓扑排序	250
5.7 关键路径	253
第 6 章 查找	258
6.1 查找的基本概念	259
6.2 线性表	260
6.3 B-树	267
6.4 B+树	275
6.5 哈希表	278
第 7 章 排序	288
7.1 排序的基本概念	290
7.2 插入排序	292
7.3 交换排序	298
7.4 选择排序	309
7.5 归并排序	318
7.6 基数排序	323
7.7 外排序	325
附录 A 数据结构部分模拟试题	330
附录 B 数据结构部分模拟试题参考答案	339
参考文献	348

第 1 章

绪 论

考研大纲

本章没有出现在考研大纲中，但是把握这些内容有助于对整个课程知识的理解。复习要求如下：

- 掌握数据结构的基本概念，包括数据的逻辑结构和存储结构，对于后面的术语要能区分哪些属于逻辑结构，哪些属于物理结构。
- 掌握算法时间和空间复杂度的分析方法。
- 理解递归算法的设计过程，掌握单链表、二叉树等递归数据结构的设计方法。

近四年全国考研题本章出题情况

年份	单项选择题/分	综合应用题/分	小计/分
2009 年	0 题	0 题	0
2010 年	0 题	0 题	0
2011 年	1 题×2	0 题	2
2012 年	1 题×2	0 题	2

1.1 数据结构的基本概念

难度: ★★
重要性: ★★★

1.1.1 知识点归纳

1. 数据

数据是指能输入到计算机中并能被计算机程序处理的符号的总称。例如, 1~100 的整数和所有大写字母都是数据。

2. 数据元素

数据元素是数据的基本单位, 作为一个整体进行处理。

注意

在数据结构课程中, 通常将数据元素简称为元素, 并误将元素和节点混用, 实际上两者是有区别的, 数据元素是逻辑概念, 在设计存储结构时, 数据元素映射成元素或节点, 如顺序存储结构中, 一个数据元素直接映射成一个元素, 在链式存储结构中, 一个数据元素映射成一个节点, 一个节点除了包含数据元素外, 还用指针表示逻辑关系。

3. 数据项

数据项是构成数据元素的不可分割的最小单位。数据元素可以由若干个数据项组成。例如, 一个学生记录就是一个数据元素, 它由学号、姓名、性别等数据项组成。

4. 数据对象

数据对象是具有相同性质的数据元素的集合, 它是数据的子集。

5. 数据结构

数据结构是相互之间存在着某种关系的数据元素的集合, 包括以下 3 个方面。

(1) 数据的逻辑结构

数据的逻辑结构是指数据元素之间逻辑关系的整体, 它与数据的存储结构无关, 是独立于计算机的。注意, 在数据结构课程中讨论的逻辑关系仅指数据元素之间的相邻关系或邻接关系。

可以采用多种方式描述数据的逻辑结构, 例如, 树可以采用树形表示、文氏图表示等描述其逻辑结构。

二元组是一种通用的逻辑结构表示方式, 它直接表示数据元素集合和数据元素之间的关系。二元组 $DS=(D, R)$, 其中 D 是数据元素的有限集, R 是 D 上关系的有限集。例如, 有如下数据 A , 即一个矩阵:

$$A = \begin{bmatrix} 2 & 6 & 3 & 1 \\ 8 & 12 & 7 & 4 \\ 5 & 10 & 9 & 11 \end{bmatrix}$$

可以用二元组表示为:

$$\begin{aligned} A &= (D, R) \\ D &= \{2, 6, 3, 1, 8, 12, 7, 4, 5, 10, 9, 11\} \\ R &= \{r_1, r_2\} \end{aligned}$$

其中 r_1 表示行关系, r_2 表示列关系:

$$\begin{aligned} r_1 &= \{ \langle 2, 6 \rangle, \langle 6, 3 \rangle, \langle 3, 1 \rangle, \langle 8, 12 \rangle, \langle 12, 7 \rangle, \langle 7, 4 \rangle, \langle 5, 10 \rangle, \langle 10, 9 \rangle, \langle 9, 11 \rangle \} \\ r_2 &= \{ \langle 2, 8 \rangle, \langle 8, 5 \rangle, \langle 6, 12 \rangle, \langle 12, 10 \rangle, \langle 3, 7 \rangle, \langle 7, 9 \rangle, \langle 1, 4 \rangle, \langle 4, 11 \rangle \} \end{aligned}$$

(2) 数据的存储结构 (物理结构)

数据的存储结构是逻辑结构用计算机语言的实现 (亦称为映像), 它是依赖于计算机语言的。设计一个逻辑结构对应的存储结构时, 不仅要存储逻辑结构中的每一个数据元素, 还要正确地表示逻辑结构中数据元素之间的逻辑关系。

例如, 上述数据 A 可用一个 C/C++ 语言中的二维数组 B 来存储, 其存储结构为:

```
int B[3][4] = {{2, 6, 3, 1}, {8, 12, 7, 4}, {5, 10, 9, 11}};
```

这种存储结构设计之所以是正确的, 是因为 B 不仅存储了 A 中所有的数据元素, 而且保存了 A 中所有元素之间的行、列关系。

(3) 施加在数据上的运算

施加在数据上的运算包括运算描述和实现, 运算描述是针对逻辑结构的, 指出运算的功能; 运算实现是针对存储结构的, 指出运算的具体操作步骤。

例如, 上述数据 A 上的一个运算为 $\text{add}(A, \&s)$, 其功能为求 A 的所有元素之和。若数据 A 采用二维数组 B 存储, 其运算实现如下:

```
void add(int B[3][4], int &s)    //s 为 B 中所有数据元素之和
{
    int i, j;
    s = 0;
    for (i = 0; i < 3; i++)
        for (j = 0; j < 4; j++)
            s += B[i][j];
}
```

6. 逻辑结构分类

根据数据元素之间的逻辑关系将逻辑结构分为以下 4 种类型。

- 集合: 数据元素之间没有任何关系。
- 线性结构: 数据元素之间存在一对一的线性关系。
- 树形结构: 数据元素之间存在一对多的层次关系。
- 图形结构: 数据元素之间存在多对多的任意关系。

7. 存储结构分类

根据数据元素在内存中的组织形式将存储结构分为以下 4 种类型。

- 顺序存储结构：把逻辑上相邻的元素存储在物理位置上相邻的存储单元里，即直接映射。
- 链式存储结构：不要求逻辑上相邻的元素在物理位置上亦相邻，元素之间的逻辑关系由附加的指针字段表示。
- 索引存储结构：在存储元素信息的同时，还建立附加的索引表。索引表中的每一项称为索引项，索引项的一般形式是：(关键字，地址)，关键字唯一标识一个元素，地址作为指向元素的指针。
- 散列（或哈希）存储结构：根据元素的关键字通过哈希函数直接计算出该元素的存储地址。

8. 抽象数据类型

抽象数据类型（ADT）是指一个数学模型以及定义在该模型上的一组操作，通常用（数据对象，数据关系，基本操作集）这样的三元组来表示，数据对象+数据关系构成数据逻辑结构。所以，ADT=逻辑结构+抽象运算（运算功能描述），可以用抽象数据类型描述一个完整的求解问题。

对于给定的 ADT，设计其存储结构，在此基础上设计实现抽象运算的算法。

1.1.2 例题解析

一、单项选择题

1. 数据结构是具有_____的数据元素的集合。【★】

- A. 性质相同
- B. 特定关系
- C. 相同运算
- D. 数据项

解：数据结构由数据元素集合和数据元素关系两部分组成，本题答案为 B。

2. 下列说法中，不正确的是_____。【★】

- A. 数据元素是数据的基本单位
- B. 数据项是数据中不可分割的最小可标识单位
- C. 数据可由若干个数据元素构成
- D. 数据项可由若干个数据元素构成

解：数据元素可由若干个数据项构成，本题答案为 D。

3. 数据的存储结构是指_____。【★】

- A. 数组类型
- B. 指针类型
- C. 数据之间的逻辑关系
- D. 数据之间的物理关系

解：数据的存储结构就是物理结构，指数据之间的物理关系。本题答案为 D。

4. 在存储数据时，通常不仅要存储各数据元素的值，而且还要存储_____。【★】

- A. 数据的处理方法
- B. 数据元素的类型
- C. 数据元素之间的关系
- D. 数据的存储方法

解：在存储数据时，需要存储数据元素的值和数据元素之间的关系。本题答案为 C。

5. 在计算机的存储器中表示时，各元素的物理地址和逻辑地址的相对顺序相同并且是连续的，称之为_____。【★】

- A. 逻辑结构
- B. 顺序存储结构
- C. 链式存储结构
- D. 以上都对

解：顺序存储结构是一种直接映射，存储结构中直接反映数据元素之间的逻辑关系。本题答案为 B。

6. 下面术语中，与数据的存储结构无关的是_____。【★★】

- A. 循环队列 B. 栈 C. 散列表 D. 单链表

解：只有栈是逻辑结构，其他选项都是存储结构。本题答案为 B。

7. 下面术语中，与数据的存储结构无关的是_____。【★★】

- A. 栈 B. 顺序栈 C. 线索二叉树 D. 双链表

解：线索二叉树是由二叉链变化过来的，属存储结构。本题答案为 A。

8. 下列各项中属于逻辑结构的是_____。【★★】

- A. 哈希表 B. 有序表 C. 单链表 D. 顺序表

解：只有有序表属于逻辑结构，其他选项都属于存储结构。本题答案为 B。

9. 以下数据结构中，_____是非线性数据结构。【★★】

- A. 栈 B. 队列 C. 完全二叉树 D. 堆

解：栈和队列都属于线性结构，堆是一个线性序列，将它看成一棵完全二叉树时满足堆的定义。本题答案为 C。

10. 对于数据结构，以下叙述中不正确的是_____。【★★】

- A. 相同的逻辑结构，对应的存储结构也必相同
B. 数据结构由逻辑结构、存储结构和基本运算三方面组成
C. 数据存储结构就是数据逻辑结构在存储器中的实现
D. 对数据基本运算的实现与存储结构有关

解：同一逻辑结构，可以设计多种对应的存储结构，基本运算的实现与存储结构有关。本题答案为 A。

11. 对于数据结构，以下叙述中不正确的是_____。【★★】

- A. 数据的逻辑结构与数据元素本身的形式和内容无关
B. 数据的逻辑结构是数据的各数据项之间的逻辑关系
C. 数据元素是数据的基本单位
D. 数据项是数据的最小单位

解：数据的逻辑结构是数据的各数据元素之间的逻辑关系。本题答案为 B。

12. 可以用_____定义一个完整的数据结构。【★★】

- A. 数据元素 B. 数据对象 C. 数据关系 D. 抽象数据类型

解：抽象数据类型描述了数据的逻辑结构和抽象运算，构成了一个完整的数据结构定义。本题答案为 D。

13. 可以用_____、数据关系和基本操作集定义一个完整的抽象数据类型。【★★】

- A. 数据元素 B. 数据对象 C. 原子类型 D. 存储结构

解：抽象数据类型可用（数据对象，数据关系，基本操作集）三元组来表示。本题答案为 B。

二、综合应用题

14. 举一个数据结构的例子, 叙述其逻辑结构、存储结构、运算三个方面的内容。【★★★】

解: 例如有一张学生成绩表, 如表 1.1 所示, 记录了一个班的学生各门课的成绩。这个表就是一个数据结构。每个记录 (包括学号, 姓名, 成绩) 就是一个节点, 对于整个表来说, 只有一个开始节点 (其前面无记录) 和一个终端节点 (其后面无记录), 其他的节点则各有一个也只有一个前趋和后继节点 (它的前面和后面均有且仅有一个记录)。这几个关系就确定了这个表的逻辑结构。

表 1.1 学生成绩表

学号	姓名	成绩
1	王华	90
2	李明	84
...	...	...

那么怎样把这个表中的数据存储到计算机里呢? 如何用高级语言表示各节点之间的关系呢? 是用一片连续的内存单元来存放这些记录 (如用数组表示), 还是随机存放各节点数据再用指针进行链接呢? 这就是存储结构的问题。例如, 若用链式存储方式, 节点的数据类型定义如下:

```
struct node
{
    int no;           //存放学号的数据域
    char name[10];    //存放姓名的数据域
    int score;        //存放成绩的数据域
    struct node *next; //指向下一个节点的指针
}
```

在建表之后就要对这张表中的记录进行查询、修改、删除等操作, 对这个表可以进行哪些操作以及如何实现这些操作就是数据运算问题了。若采用链式存储方式, 其运算的实现请参见第 2 章中单链表的相关算法。

15. 数据运算是数据结构的一个重要方面。试举一例说明两个数据结构的逻辑结构和存储方式完全相同, 只是对于运算的定义不同, 因而两个数据结构具有显著不同的特性, 构成两个不同的数据结构。【★★★】

解: 在数据结构中这类例子较多, 如顺序表和字符串等。下面以二叉树和二叉排序树进行说明。

二叉树的定义为: 二叉树是有限的节点集合, 这个集合或者是空, 或者由一个根节点和两棵互不相交的称为左子树和右子树的二叉树组成。

二叉排序树的定义为: 二叉排序树或者是空树, 或者是满足如下性质的二叉树:

- 若它的左子树非空, 则左子树上所有记录的值均小于根记录的值。
- 若它的右子树非空, 则右子树上所有记录的值均大于根记录的值。
- 左、右子树本身又各是一棵二叉排序树。

两者的逻辑结构和存储结构完全相同, 因为二叉排序树可以采用二叉树的逻辑表示和存储方式。两者的运算有建立树、插入节点、删除节点、查找节点等。对于二叉树和二叉排序树, 这些运算的定义是不同的, 以查找节点为例, 二叉树的时间复杂度为 $O(n)$, 而二叉排序树的时间复杂度为 $O(\log_2 n)$ 。

二叉树和二叉排序树是两个不同的数据结构。前者通常用于表示层次关系，后者通常用于排序和查找。

1.2 算法及其分析

难度: ★★★

重要性: ★★★

1.2.1 知识点归纳

1. 什么是算法

算法是对特定问题求解步骤的一种描述，它是指令的有限序列。

2. 算法的特性

算法的5个重要特性如下。

- 有穷性：算法必须总是（对任何合法的输入值）在执行有穷步之后结束。
- 确定性：算法无二义性，对于同一种输入只能有同一种结果。
- 可行性：算法中的所有操作必须通过已经实现的基本运算有限次实现。
- 有输入：包含一个或多个输入。
- 有输出：包含一个或多个输出。

注意

算法和程序的区别在于有穷性，程序不一定具有有穷性，例如，Windows 操作系统可以不停机（在硬件设备不出现故障时可以长期运行）。

3. 算法的时间复杂度

算法时间复杂度是算法中基本运算重复执行次数多少的量度。算法中基本运算次数 $T(n)$ 是问题规模 n （输入量的多少，称之为问题规模）的某个函数 $f(n)$ ，记作：

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

表示当 $n \geq n_0$ （ n_0 为一个尽可能大的正整数）， $T(n)=cf(n)$ ， c 为正数。

记号“O”读作“大O”，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。

对于给定的算法，求时间复杂度的一般过程是：首先确定问题规模 n ，再选取算法中循环最深的语句作为基本运算，求出其执行次数 $T(n)$ ，最后提取出最高阶并置其序数为1，从而得到 $f(n)$ 。对于递归算法，需总结出递归模型，推导出 $T(n)$ ，再导出 $f(n)$ 。

4. 算法的空间复杂度

算法空间复杂度是对一个算法在运行过程中临时占用的存储空间大小的量度。所谓临时空间，对于用一个函数实现的算法，仅考虑函数体内开辟的空间，而不需要考察形参的空间。空间复杂度也和时间复杂度一样用 O 表示， $S(n)=O(g(n))$ 。

对于给定的算法，求空间复杂度的一般过程是：首先确定问题规模 n ，再选取算法函数体中定义的变量个数或用 `malloc` 函数开辟的空间个数，求出其临时空间量 $S(n)$ ，最后提取出最高阶并置其序数为 1，从而得到 $g(n)$ 。对于递归算法，需总结出递归模型，推导出 $S(n)$ ，再导出 $g(n)$ 。

空间复杂度为 $O(1)$ 的算法称为原地工作或就地工作算法。

1.2.2 例题解析

一、单项选择题

1. ____ 不是算法的基本特性。【★】

- A. 可行性 B. 长度有限 C. 在规定的时间内完成 D. 确定性

解：选项 C 指的是有穷性。而有穷性并不是指长度有限，而是指执行的时间是有限的。本题答案为 B。

2. 某算法的时间复杂度为 $O(n^2)$ ，表明该算法的 ____。【★★】

- A. 问题规模是 n^2 B. 执行时间等于 n^2
C. 执行时间与 n^2 成正比 D. 问题规模与 n^2 成正比

解： $T(n)=O(n^2)$ 表示 $T(n)=m \times n^2$ (m 为正常量)，其问题规模仍为 n 而不是 n^2 。本题答案为 C。

注意

算法时间复杂度是问题规模 n 的函数，记为 $T(n)=O(f(n))$ ，表示 $T(n)=cf(n)$ ，其中 c 为正常量，所以， $T(n)$ 的增长率与 $f(n)$ 的增长率相同。

3. 一个算法的执行时间为 $T(n)=(3n^2+2n\log_2 n+4n-7)/(10n)$ ，其时间复杂度为 ____。【★★】

- A. $O(3n^2)$ B. $O(2n\log_2 n)$ C. $O(3n/10)$ D. $O(n)$

解：当 n 足够大时， $T(n) \rightarrow 3n/10 = O(n)$ 。本题答案为 D。

4. 以下有关算法的说法错误的是 ____。【★★★】

- I. 算法原地工作的含义是指不需要任何额外的辅助空间
II. 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2n)$ 的算法
III. 所谓最坏时间复杂度是指最坏情况下估算算法执行时间的一个上界
IV. 同一个算法，实现语言的级别越高，执行效率就越低
A. I B. I 和 II C. I 和 IV D. III

解：算法原地工作的含义是指算法的空间复杂度为 $O(1)$ ，同一个算法实现语言的级别越高，执行效率并不一定越低。本题答案为 C。

5. 求整数 n ($n \geq 0$) 阶乘的算法如下，其时间复杂度是 ____。【★★★】

```
int fact(int n)
{
    if (n <= 1) return 1;
    else return n * fact(n-1);
}
```