

普通高等院校计算机类专业规划教材·精品系列

实用数据结构基础 学习指导

SHIYONG SHUJU JIEGOU JICHU XUEXI ZHIDAO

(第二版)

陈元春 王淮亭 王中华 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校计算机类专业规划教材·精品系列

实用数据结构基础 学习指导

(第二版)

陈元春 王淮亭 王中华 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

本书是《实用数据结构基础（第四版）》（陈元春、王中华、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5部分：教学内容指导（包括对主教材第1~10章的知识点分析、典型习题分析和各章习题解答）；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指导）；模拟试卷；模拟试卷参考答案；数据结构课程设计报告样例。

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。

本书适合作为普通高等院校计算机类专业数据结构课程的教学辅导书，也可作为成人教育、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, 王中华
编著. — 2版. — 北京: 中国铁道出版社, 2015.9
普通高等院校计算机类专业规划教材. 精品系列
ISBN 978-7-113-20775-5

I. ①实… II. ①陈… ②王… ③王… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆CIP数据核字(2015)第173726号

书 名: 实用数据结构基础学习指导（第二版）

作 者: 陈元春 王淮亭 王中华 编著

策 划: 周海燕

读者热线: 400-668-0820

责任编辑: 周海燕 彭立辉

封面设计: 穆 丽

封面制作: 白 雪

责任校对: 汤淑梅

责任印制: 李 佳

出版发行: 中国铁道出版社（100054，北京市西城区右安门西街8号）

网 址: <http://www.51eds.com>

印 刷: 北京华正印刷有限公司

版 次: 2008年11月第1版 2015年9月第2版 2015年9月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 13 字数: 318千

书 号: ISBN 978-7-113-20775-5

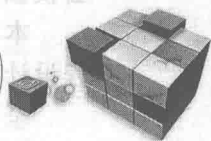
定 价: 28.00元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社教材图书营销部联系调换。电话：（010）63550836

打击盗版举报电话：（010）51873659

前言（第二版）



数据结构是计算机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方法。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应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，初学者不容易理解和掌握，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（第四版）》（陈元春、王中华、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 5 部分。

第 1 部分 教学内容指导，包括对主教材第 1~10 章的知识点分析、典型习题分析和各章习题解答。

第 2 部分 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指导，并给出了完整的 C 程序源代码。所有源程序都已在 Visual C++ 6.0 环境下运行通过。通过这些实验，学生可以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 3 部分 模拟试卷，包含了 2 套模拟试卷，主要是为了帮助学生在学完数据结构课程以后自我检验。每套试卷中有判断题、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，其目的是帮助学生学完数据结构课程后进行系统复习和自我测试。

第 4 部分 模拟试卷参考答案，供学生参考。

第 5 部分 课程设计报告样例，为读者提供了“设计并实现大整数的加减乘除运算”的设计报告例子，供学生参考。

由于《实用数据结构基础（第四版）》对各章的习题进行了改写和充实，所以本指导书的第 1 部分也做了相应的修订。除了对主教材第 1~10 章的知识点、典型习题进行分析以外，还提供了教材各章习题的全部解答。对第 2 部分内容只做了少量的修改。这两部分内容仍然由陈元春和王淮亭编写。

本次修订对原书的第 3、第 4 部分模拟试卷及参考答案进行了重写，并把原来 6

套模拟试卷压缩为两套，这部分内容仍然由陈元春编写。

本次修订增加了第5部分课程设计报告样例，这部分内容由王中华根据学生的课程设计报告修改、整理编写，主要用来指导学生如何撰写数据结构课程设计报告。

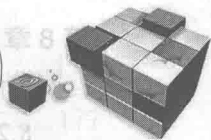
全书由陈元春定稿。本书适合作为《实用数据结构基础（第四版）》的教学指导用书，也可以作为数据结构自学者的参考书。

由于编者水平有限，加之成书时间仓促，书中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

2015年6月

目 录



第 1 部分 教学内容指导

第 1 章 绪论	2
1.1 知识点分析	2
1.2 典型习题分析	3
1.3 习题 1 解答	6
第 2 章 线性表	8
2.1 知识点分析	8
2.2 典型习题分析	9
2.3 习题 2 解答	15
第 3 章 栈	19
3.1 知识点分析	19
3.2 典型习题分析	21
3.3 习题 3 解答	24
第 4 章 队列	29
4.1 知识点分析	29
4.2 典型习题分析	30
4.3 习题 4 解答	34
第 5 章 串	39
5.1 知识点分析	39
5.2 典型习题分析	40
5.3 习题 5 解答	44
第 6 章 多维数组和广义表	49
6.1 知识点分析	49
6.2 典型习题分析	50
6.3 习题 6 解答	54
第 7 章 树和二叉树	57
7.1 知识点分析	57
7.2 典型习题分析	59



7.3 习题 7 解答.....	64
第 8 章 图	75
8.1 知识点分析.....	75
8.2 典型习题分析.....	76
8.3 习题 8 解答.....	80
第 9 章 查找	87
9.1 知识点分析.....	87
9.2 典型习题分析.....	88
9.3 习题 9 解答.....	95
第 10 章 排序	101
10.1 知识点分析.....	101
10.2 典型习题分析.....	103
10.3 习题 10 解答.....	107

第 2 部分 自主设计实验指导

实验 1 学生成绩分析程序.....	113
实验 2 多项式求和.....	118
实验 3 后缀表达式求值.....	123
实验 4 循环队列的实现和运算.....	129
实验 5 字符串分割处理.....	133
实验 6 稀疏矩阵十字链表的存储.....	137
实验 7 标识符树与表达式求值.....	144
实验 8 最小生成树.....	147
实验 9 哈希查找.....	150
实验 10 双向冒泡排序.....	155

第 3 部分 模拟试卷

模拟试卷 1.....	158
模拟试卷 2.....	163

第 4 部分 模拟试卷参考答案

模拟试卷 1 参考答案.....	169
模拟试卷 2 参考答案.....	172

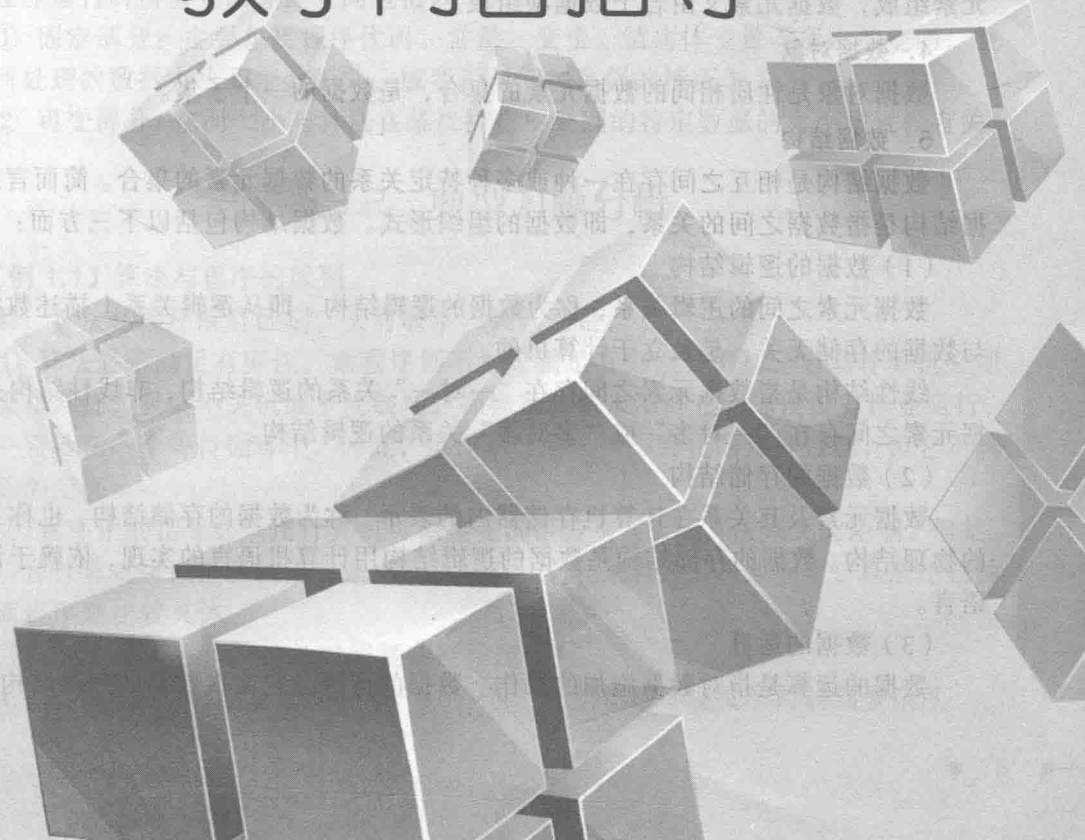
第 5 部分 课程设计报告样例

1 课题概述.....	177
1.1 课题的目的.....	177
1.2 课题的要求.....	177
1.2.1 输入/输出的要求.....	177
1.2.2 程序实现的功能要求.....	177
2 概要设计.....	178
2.1 程序的模块结构.....	178
2.2 大整数存储结构的选择.....	178
2.3 输入数据合法性的检测及初步处理.....	179
2.4 算法的描述.....	179
2.4.1 加法运算.....	179
2.4.2 减法运算.....	180
2.4.3 乘法运算.....	181
2.4.4 除法运算.....	182
3 程序功能的实现.....	187
3.1 主函数的实现.....	187
3.2 主要功能模块的实现.....	188
3.2.1 加法的实现.....	188
3.2.2 减法的实现.....	189
3.2.3 乘法的实现.....	191
3.2.4 除法的实现.....	192
4 程序调试及问题解决.....	195
5 程序测试及分析.....	196
6 总结.....	198
参考文献.....	200

»» 目录

第 1 部分

教学内容指导





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。

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 3 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。

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。

(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,当一个算法比较复杂时,其时间复杂度的计算会变得相当困难。一般情况下,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 $T(n)$ 的数量级可记作 $T(n)=O(f(n))$ 。

算法的时间复杂度 $T(n)$ 是该算法的时间消耗,一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,是算法中最大的语句频度。

一般情况下,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

一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,记作 $S(n)=O(f(n))$ 。其中, n 为问题的规模。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,当用一台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执行时间的上界, 所以应选 B。

解: B。

【例 1.3】 当 n 取 $1 \sim 10$ 时, 比较 n 、 $2n$ 、 n^2 、 n^3 、 2^n 、 $n!$ 、 n^n 增长率的变化。

解: 表 1-1-1 所示为当 n 取 $1 \sim 10$ 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 各函数值增长率

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 $T_4(n)=2n^2+6000n\lg n$ 。

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 $T_1(n)=5n^2+10n+6\lg n=5n^2+O(n)$, $T_2(n)=3n^2+100n+3\lg n=3n^2+O(n)$,

$T_3(n)=8n^2+3lgn=8n^2+O(lgn)$, $T_4(n)=2n^2+6000nlgn=2n^2+O(nlgn)$ 。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 3 个函数 $f(n)$ 、 $g(n)$ 、 $h(n)$ 分别为: $f(n)=100n^3+n^2+100$ 、 $g(n)=25n^3+50n^2$ 、 $h(n)=n^{1.5}+50nlgn$, 判断下列关系是否成立:

- ① $f(n)=O(g(n))$ ② $h(n)=O(n^{1.5})$ ③ $h(n)=O(nlgn)$

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 $nlgn$, 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 lgn , 也大于 $nlgn$, 即 $\lim h(n)/n^{1.5}=\lim(n^{1.5}+50nlgn)/n^{1.5}=1$, 所以② $h(n)=O(n^{1.5})$ 成立。

③ $\lim h(n)/nlgn=\lim(n^{1.5}+50nlgn)/nlgn=\infty$, 所以③ $h(n)=O(nlgn)$ 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 习题 1 解答

一、判断题答案

题目	(1)	(2)	(3)	(4)	(5)
答案	×	√	√	×	√

二、填空题答案

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

三、选择题答案

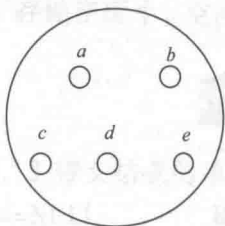
题目	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
答案	B	C	A	D	B	B	A	B	D	C
题目	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
答案	A	C	C	A	D	D	C	C	B	D

四、答案

- (1) $O(n \times m)$ (2) $O(n^2)$ (3) $O(1)$
 (4) $O(n)$ (5) $O(n^2)$ (6) $O(m \times n \times l)$

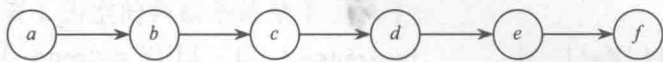
五、答案

(1)



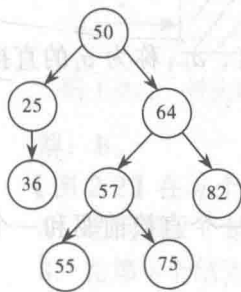
属于集合

(2)



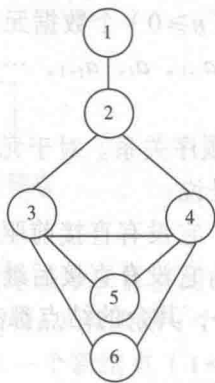
属于线性结构

(3)



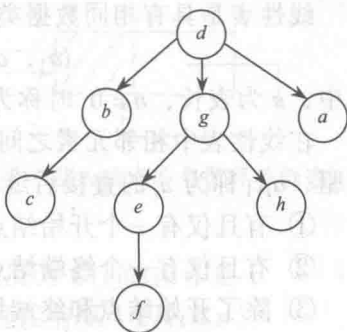
属于树形结构

(4)



属于图形结构

(5)



属于树形结构

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。

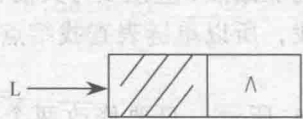


图 1-2-1 带头结点的空单链表

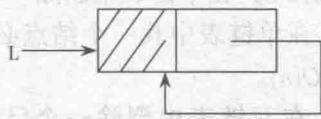


图 1-2-2 带头结点的空循环单链表

解: B。

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