



名师考案 丛书
MINGSHI KAOAN CONGSHU

(清华·C语言版)

数据结构

考研教案

夏清国 主编

西北工业大学出版社



名师考案丛书

MINGSHIKAOANCONGSHU

TP311.12
100C

数据结构考研教案

(清华·C语言版)

主编 夏清国

编者 姚群 蔡康英 李联 刘军兰

西北工业大学出版社

【内容简介】 本书是根据严蔚敏、吴伟民编著的《数据结构》(C语言版)而编写的辅助教材。

全书共分两部分,第1部分按原教材章节顺序,分为12章,每章按知识脉络图解、重点和难点解读、课程考试和考研要点、典型例题及习题精选详解等四个知识模块编写。第2部分给出了几套课程考试模拟试题和精选了部分各大名校近几年硕士研究生入学考试题,并附有参考答案。本书内容充实完整,层次分明,概念清晰,表达方式易于接受,有利于学生对“数据结构”课程知识的理解,帮助学生从广度和深度上把握知识体系,拓宽解题思路,提高解题能力。

本书针对性强、习题覆盖面广,适用于考研者应试提高,也可作为计算机及相关专业学生学习“数据结构”课程的指导书。

图书在版编目(CIP)数据

数据结构考研教案/夏清国主编.——西安:西北工业大学出版社,2006.3

(名师考案丛书)

ISBN 7-5612-1946-6

I. 数… II. 夏… III. 数据结构—研究生—入学考试—自学参考资料 IV. TP311.12

中国版本图书馆CIP数据核字(2005)第134182号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路127号 邮编:710072

电话:029-88493844 88491757

网址:www.nwpu.com

印刷者:西安东江印务有限公司

开本:787mm×960mm 1/16

印张:28.125

字数:751千字

版次:2006年3月第1版 2006年3月第1次印刷

印数:1~6000册

定价:36.00元



前 言

“数据结构”课程是计算机专业、信息管理专业和其他一些与计算机技术关系密切专业的一门必修的核心课程，对初学者来说是比较困难和抽象的一门课程。为了满足广大读者学习的需求，帮助广大读者学好“数据结构”课程，从容应对考试，我们根据多年的教学经验编写了本书。

本书是配合清华大学出版社出版的严蔚敏、吴伟民编著的《数据结构》(C语言版)所编写的辅助教材，共分两大部分：

第一部分，按原教材章节顺序，分为12章，每章按四个知识模块编写：

一、知识脉络图解：用框图形式简捷集中地列出各章节的知识点。

二、重点和难点解读：高度概括章节的要点。

三、课程考试和考研要点：列出了本章应掌握的考试内容和知识要点。

四、典型例题及习题精选详解：根据数据结构课程的教学规律，总结出每章的考点，精选出常见的典型题目，进行详细分析解答。同时提供了大量习题和参考答案，以使读者检查自己的学习效果和应试能力，辅助读者进一步加深对所学内容的理解及增强解题能力。

第二部分，课程考试模拟试题及参考答案和考研真题及解答，以供读者了解课程考试与考研的题量和模式，进行实战演练。

本书从指导课程教学、学习和考试的角度出发，通过对大量常见经典题型进行分析，给出了“数据结构”的解题方法、解题规律和解题技巧。这对于提高读者分析问题的能力，理解基本要领和理论，开拓解题思路，将会起到良好的效果。对于课程考试模拟试题及考研真题，希望读者在学习过程中先独立思考，自己动手解题，然后再对照参考答案检查，不要依赖于答案。

由于作者水平有限，书中疏漏与不妥之处，恳请读者批评指正。

编 者

2005年7月



目 录

第一部分

第 1 章 绪论	1
1.1 知识脉络图解	1
1.2 重点、难点解读	1
1.3 课程考试、考研要点点击	1
1.4 典型例题及习题精选详解	3
第 2 章 线性表	15
2.1 知识脉络图解.....	15
2.2 重点、难点解读.....	15
2.3 课程考试、考研要点点击.....	15
2.4 典型例题及习题精选详解.....	23
第 3 章 栈和队列	60
3.1 知识脉络图解.....	60
3.2 重点、难点解读.....	60
3.3 课程考试、考研要点点击.....	60
3.4 典型例题及习题精选详解.....	66
第 4 章 串	93
4.1 知识脉络图解.....	93
4.2 重点、难点解读.....	93
4.3 课程考试、考研要点点击.....	93



4.4 典型例题及习题精选详解	104
第5章 数组和广义表	122
5.1 知识脉络图解	122
5.2 重点、难点解读	122
5.3 课程考试、考研要点点击	122
5.4 典型例题及习题精选详解	133
第6章 树和二叉树	160
6.1 知识脉络图解	160
6.2 重点、难点解读	160
6.3 课程考试、考研要点点击	160
6.4 典型例题及习题精选详解	171
第7章 图	210
7.1 知识脉络图解	210
7.2 重点、难点解读	210
7.3 课程考试、考研要点点击	210
7.4 典型例题及习题精选详解	218
第8章 动态存储管理	263
8.1 知识脉络图解	263
8.2 重点、难点解读	263
8.3 课程考试、考研要点点击	263
8.4 典型例题及习题精选详解	267
第9章 查找	276
9.1 知识脉络图解	276
9.2 重点、难点解读	276
9.3 课程考试、考研要点点击	276
9.4 典型例题及习题精选详解	284
第10章 内部排序	316
10.1 知识脉络图解	316





10.2	重点、难点解读	316
10.3	课程考试、考研要点点击	316
10.4	典型例题及习题精选详解	324
第 11 章	外部排序	359
11.1	知识脉络图解	359
11.2	重点、难点解读	359
11.3	课程考试、考研要点点击	359
11.4	典型例题及习题精选详解	361
第 12 章	文件	366
12.1	知识脉络图解	366
12.2	重点、难点解读	366
12.3	课程考试、考研要点点击	366
12.4	典型例题及习题精选详解	369
第二部分		
I.	课程考试模拟试题及参考答案	378
II.	考研真题及参考答案	405



第一部分

第1章 绪论

1.1 知识脉络图解

绪论 { 基本概念(数据、数据元素、数据对象、数据结构)
数据的逻辑结构(集合、线性表、树形结构、图状结构)
数据的存储结构(线性存储结构、链式、索引、散列存储结构)
数据类型(原子类型、结构类型、抽象数据类型)
算法(什么是算法、算法具有的特性)
算法效率的度量(时间复杂度、空间复杂度)

1.2 重点、难点解读

重点:数据结构的逻辑结构、存储结构、数据运算三方面的概念及相互关系;算法的时间复杂度分析。
难点:分析算法的时间复杂度。

1.3 课程考试、考研要点点击

1.3.1 基本概念和术语

(1) 数据:数据是对客观事物的符号表示,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。



(2) 数据元素:数据元素是数据的基本单位,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。

(3) 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。

(4) 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。数据结构包括三方面的内容:数据的逻辑结构、数据的存储结构和数据的运算。

(5) 数据的逻辑结构:

1) 集合:结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系。

2) 线性结构:结构中的数据元素之间存在一对一的关系。若结构是非空集,则有且仅有一个开始结点和一个终端结点,并且除开始结点无直接前趋和终端结点无直接后继外,其他所有结点都只有一个直接前趋和一个直接后继。

3) 树形结构:结构中的数据元素之间存在一对多的关系。若结构是非空集,则除第一个结点外,其他所有结点都只有一个直接前趋,除叶子结点外,其他所有结点可能有多个直接后继。

4) 图状结构:结构中的数据元素之间存在多对多关系。若结构是非空集,所有结点都可能多个直接前趋和多个直接后继。

(6) 数据的存储结构:

1) 顺序存储方法。该方法是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邻接关系来体现,由此得到的存储结构称为顺序存储结构,通常顺序存储结构是借助于程序语言的数组来描述的。

2) 链接存储方法。该方法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由附加的指针字段表示的,由此得到的存储表示称为链式存储结构,通常要借助于程序语言的指针类型来描述它。

3) 索引存储方法。该方法通常是在存储结点信息的同时,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)。其中关键字唯一标识结点,地址作为指向结点的指针。

4) 散列存储方法。该方法的基本思想是根据结点的关键字直接计算出该结点的存储地址。

(7) 数据的运算:数据的运算是在数据的逻辑结构上定义的操作算法,如检索、插入、删除、更新和排序等。

(8) 数据类型:

1) 原子类型:其值不可再分的数据类型。

2) 结构类型:其值可以再分解为若干成分(分量)的数据类型。

3) 抽象数据类型:抽象数据组织和与之相关的操作。

1.3.2 算法和算法分析

1. 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,其中每条指令表示一个或多个操作。

• 特性:

(1) 有穷性:一个算法必须总是(对任何合法的输入)在执行有穷步之后结束,且每一步都可在有穷时间内完成。

(2) 确定性:算法中每一条指令必须有确切的含义,确保不会产生二义性。并且,在任何条件下,算法只



有唯一的一条执行路径,即对于相同的输入只能得出相同的输出。

(3) 可行性:一个算法是可行的,即算法中描述的操作都是可以通过已实现的基本运算执行有限次来实现的。

(4) 输入性:一个算法有 0 个或多个的输入。

(5) 输出性:一个算法有一个或多个的输出。

2. 算法效率的度量

(1) 时间复杂度。一个语句的频度,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度之和记作 $T(n)$,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, $T(n)$ 的数量级称为渐近时间复杂度,简称为时间复杂度,记作 $T(n) = O(f(n))$ 。

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状态。一个问题的输入实例是满足问题陈述中所给出的限制和为计算该问题的解所需要的所有输入构成的。

最坏时间复杂度是指在最坏情况下算法的时间复杂度。

平均时间复杂度是指所有可能的输入实例均以等概率出现的情况下,算法的期望运行时间。

上述表达式中“ O ”的含义是 $T(n)$ 的数量级,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 $0 \leq T(n) \leq C \times f(n)$ 。

一般总是考虑在最坏的情况下的时间复杂度,以保证算法的运行时间不会比它更长。

(2) 空间复杂度。算法的空间复杂度 $S(n)$,定义为该算法所耗费的存储空间,它是问题规模 n 的函数。渐近空间复杂度也常常简称为空间复杂度,记作 $S(n) = O(f(n))$ 。

1.4 典型例题及习题精选详解

1.4.1 典型例题

例题 1.1 简述算法的确定性的含义。

【例题解答】 算法的确定性是指算法中每一条指令必须有确切的含义,读这些指令时不会产生二义性;并且在任何条件下,算法只有唯一的一条执行路径,即对于相同的输入只能得到相同的输出。

例题 1.2 常用的存储表示方法有几种?

【例题解答】 常用的存储表示方法有 4 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,由此得到的存储结构称为顺序存储结构。

(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,结点之间的逻辑关系是由附加的指针字段表示的,由此得到的存储结构称为链式存储结构。

(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。

(4) 散列存储方法:根据结点的关键字直接计算出该结点的存储地址。

例题 1.3 逻辑结构和存储结构之间有什么关系?

【例题解答】 对于已经建立的逻辑结构是设计人员根据解题需要选定的数据组织形式,因此建立的机



内表示应遵循选定的逻辑结构,所建立数据的机内表示称为数据存储结构。

例题 1.4 设有数据逻辑结构为: $line = (D,R)$,其中

$$D = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{r\}$$

$$r = \{ \langle 05,01 \rangle, \langle 01,03 \rangle, \langle 03,08 \rangle, \langle 08,02 \rangle, \langle 02,07 \rangle, \langle 07,04 \rangle, \langle 04,06 \rangle, \langle 06,09 \rangle, \langle 09,10 \rangle \}$$

试分析该数据结构属于哪种逻辑结构。

【例题解答】 对应的图形如图 1.1 所示。

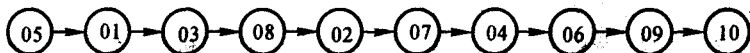


图 1.1 数据的线性结构示意图

在 $line$ 中,每个数据元素有且仅有一个直接前趋元素(除结构中第一个元素 05 外),有且仅有一个直接后继元素(除结构中最后一个元素 10 外)。这种数据结构的特点是数据元素之间的 1 对 1(1:1) 关系,即线性关系,因此本题所给定的数据结构为线性结构。

例题 1.5 设有数据逻辑结构为: $tree = (D,R)$,其中

$$D = \{01,02,03,04,05,06,07,08,09,10\}$$

$$R = \{r\}$$

$$r = \{ \langle 01,02 \rangle, \langle 01,03 \rangle, \langle 01,04 \rangle, \langle 02,05 \rangle, \langle 02,06 \rangle, \langle 03,07 \rangle, \langle 03,08 \rangle, \langle 03,09 \rangle, \langle 04,10 \rangle \}$$

试分析该数据结构属于哪种逻辑结构。

【例题解答】 对应的图形如图 1.2 所示。

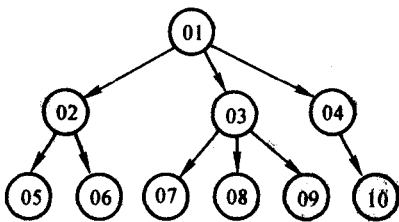


图 1.2 数据的树形结构示意图

图 1.2 像倒着画的一棵树,在这棵树中,最上面的一个结点没有前趋只有后继,称做树根结点,最下面一层的结点只有前趋没有后继,称做树叶结点。在一棵树中,每个结点有且只有一个前趋结点(除树根结点外),但可以有任意多个后继结点(树叶结点可看做具有 0 个后继结点)。这种数据结构的特点是数据元素之间的 1 对 $N(1:N)$ 关系($N \geq 0$),即层次关系,因此本题所给定的数据结构为树形结构。

例题 1.6 设有数据逻辑结构为: $graph = (D,R)$,其中

$$D = \{01,02,03,04,05,06,07\}$$



$$R = \{r\}$$

$$r = \{ \langle 01, 02 \rangle, \langle 02, 01 \rangle, \langle 01, 04 \rangle, \langle 04, 01 \rangle, \langle 02, 03 \rangle, \langle 03, 02 \rangle, \langle 02, 06 \rangle, \langle 06, 02 \rangle, \langle 02, 07 \rangle, \langle 07, 02 \rangle, \langle 03, 07 \rangle, \langle 07, 03 \rangle, \langle 04, 06 \rangle, \langle 06, 04 \rangle, \langle 05, 07 \rangle, \langle 07, 05 \rangle \}$$

试分析该数据结构属于哪种逻辑结构。

【例题解答】 对应的图形如图 1.3 所示。

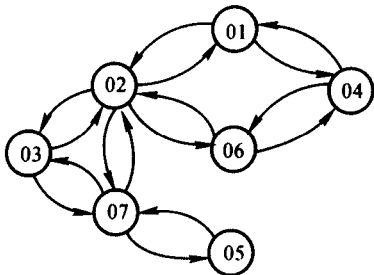


图 1.3 数据的图形结构示意图

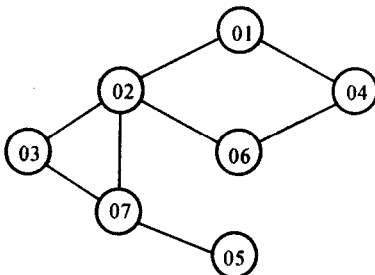


图 1.4 图 1.3 的等价表示

从图 1.3 可以看出, r 是 D 上的对称关系, 为了简化起见, 把 $\langle x, y \rangle$ 和 $\langle y, x \rangle$ 这两个对称序偶用一个无序对 (x, y) 或 (y, x) 来代替; 在图形表示中, 把 x 结点和 y 结点之间两条相反的有向边用一条无向边来代替. 这样 r 关系可改写为

$$r = \{(01, 02), (01, 04), (02, 03), (02, 06), (02, 07), (03, 07), (04, 06), (05, 07)\}$$

对应的图形如图 1.4 所示。

从图 1.3 或图 1.4 可以看出, 结点之间的联系是 M 对 N ($M:N$) 联系 ($M \geq 0, N \geq 0$), 即网状关系. 也就是说, 每个结点可以有多个前趋结点和多个后继结点. 因此本题所给定的数据结构为网状结构。

例题 1.7 设有数据逻辑结构为: $B = (K, R)$, 其中

$$K = \{k_1, k_2, k_3, k_4, k_5, k_6\}$$

$$R = \{r_1, r_2\}$$

$$r_1 = \{ \langle k_3, k_2 \rangle, \langle k_3, k_5 \rangle, \langle k_2, k_1 \rangle, \langle k_5, k_4 \rangle, \langle k_5, k_6 \rangle \}$$

$$r_2 = \{ \langle k_1, k_2 \rangle, \langle k_2, k_3 \rangle, \langle k_3, k_4 \rangle, \langle k_4, k_5 \rangle, \langle k_5, k_6 \rangle \}$$

试画出 B 对应的图形, 并分析其特征。

【例题解答】 若用实线表示关系 r_1 , 虚线表示关系 r_2 , 则对应的图形如图 1.5 所示。

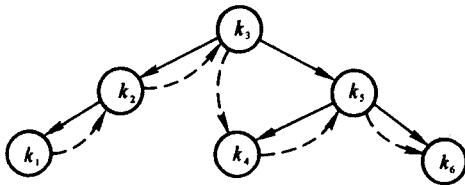


图 1.5 带有两个关系的一种数据结构示意图



从图 1.5 中可以看出,数据结构 B 是一种非线性的图形结构。但是,若只考虑关系 r_1 则为树形结构,若只考虑关系 r_2 则为线性结构。

例题 1.8 设 n 为整数,指出下列各算法的时间复杂度。

(1) void prime(int n) // n 为一个正整数

```
{ int i = 2;
  while((n%i) != 0 && i * 1.0 < sqrt(n))
    i++;
  if(i * 1.0 > sqrt(n))
    printf("%d 是一个素数\n", n);
  else
    printf("%d 不是一个素数\n", n);
}
```

(2) sum1(int n) // n 为一个正整数

```
{ int p = 1, sum = 0, i;
  for(i = 1; i <= n; i++)
  { p * = i;
    sum += p;
  }
  return sum;
}
```

(3) sum2(int n) // n 为一个正整数

```
{ int sum = 0, i, j;
  for(i = 1; i <= n; i++)
  { p = 1;
    for(j = 1; j <= i; j++)
      p * = j;
    sum += p;
  }
  return sum;
}
```

【例题解答】 (1) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。prime 算法的嵌套最深层语句为

$i++$;

它的执行次数由条件 $(n \% i) != 0 \ \&\& \ i * 1.0 < \text{sqrt}(n)$ 决定,显然执行次数小于 $\text{sqrt}(n)$,所以 prime 算法的时间复杂度是 $O(n^{1/2})$ 。

(2) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。sum1 算法的嵌套最深层语句为

$p * = i$;



$$\text{sum} += p;$$

它的执行次数为 n 次, 所以 sum1 算法的时间复杂度是 $O(n)$ 。

(3) 算法的时间复杂度是由嵌套最深层语句的执行次数决定的。 sum2 算法的嵌套最深层语句为

$$p * = j;$$

它的执行次数为 $1 + 2 + 3 + \dots + n = n(n+1)/2$ 次, 所以 sum2 算法的时间复杂度是 $O(n^2)$ 。

例题 1.9 将数量级 $O(1), O(n), O(n^2), O(n^3), O(n \ln n), O(\ln n), O(2^n)$ 按增长率从小到大排列。

【例题解答】 在题目给出的 7 种类型的数量级中, $O(1)$ 为常量型, $O(n)$ 为线性型, $O(n^2)$ 为平方型, $O(n^3)$ 为立方型, $O(n \ln n)$ 为线性对数型, $O(\ln n)$ 为对数型, $O(2^n)$ 为指数型。这 7 种类型按增长率从小到大排列如下:

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

例题 1.10 已知有实现同一功能的两个算法, 其时间复杂度分别为 $O(2^n)$ 和 $O(n^{10})$, 假设现实计算机可连续运算的时间为 10^7 s (100 多天), 又每秒可执行基本操作 (根据这些操作来估算算法时间复杂度) 10^5 次。试问在此条件下, 这两个算法可解问题的规模 (即 n 值的范围) 各为多少? 哪个算法更适宜?

【例题解答】 两个算法可解问题的规模各为 $n \leq \lg 10^{12}$ 和 $n^{10} \leq 10^{12}$ 。结论是第一个算法较适宜。由此可见, 虽然一般情况下多项式阶的算法优于指数阶的算法, 但高次多项式的算法在 n 的很大范围内不如某些指数阶的算法。

1.4.2 习题精选详解

一、单项选择题

1. 计算机算法必须具备输入、输出、() 等 5 个特性。

- A. 可行性、可移植性和可扩展性 B. 可行性、确定性和有穷性
C. 确定性、有穷性和稳定性 D. 易读性、安全性和稳定性

【解答】 B

2. 设三个函数 f, g, h 分别为 ()。

$$f(n) = 100n^3 + n^2 + 1000$$

$$g(n) = 25n^3 + 4000n^2$$

$$h(n) = n^{1.01} + 1000n \ln n$$

以下关系式中错误的是 ()。

- A. $f(n) = O(g(n))$ B. $g(n) = O(f(n))$
C. $h(n) = O(n^{1.01})$ D. $h(n) = O(n \ln n)$

【解答】 D

3. 下面程序段的时间复杂性的量级为 ()。

```
for(i = 1; i <= n; i++)
```

```
    for(j = 1; j <= i; j++)
```

```
        for(k = 1; k <= j; k++)
```

```
            x = x + 1;
```

- A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n^3)$



【解答】D

4. 数据结构是一门研究非数值计算的程序设计问题中计算机的(①)以及它们之间的(②)和运算等的学科。

- ① A. 数据元素 B. 计算方法 C. 逻辑存储 D. 数据映像
 ② A. 结构 B. 关系 C. 运算 D. 算法

【解答】①A. ②B

5. 在数据结构中,从逻辑上可以把数据结构分成()。

- A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
 C. 线性结构和非线性结构 D. 内部结构和外部结构

【解答】C

6. 下述函数中渐近时间最小的是()。

- A. $T1(n) = n \lg n + 1\ 000 \lg n$ B. $T2(n) = n^{\lg n} - 1\ 000 \lg n$
 C. $T3(n) = n^2 - 1\ 000 \lg n$ D. $T4(n) = 2n \lg n - 1\ 000 \lg n$

【解答】A

7. 下面()的时间复杂性最好,即执行时间最短。

- A. $O(n)$ B. $O(\lg n)$ C. $O(n \lg n)$ D. $O(n^2)$

【解答】B

8. 下面程序段的时间复杂性的量级为()。

```
int fun(int n)
{ int i = 1, s = 1;
  while(s < n)
    s += ++i;
  return i;
}
```

- A. $O(n/2)$ B. $O(\lg n)$ C. $O(n)$ D. $O(\sqrt{n})$

【解答】D

9. 下面程序段的时间复杂性的量级为()。

```
for(int i = 0; i < m; i++)
  for(int j = 0; j < n; j++)
    a[i][j] = i * j;
```

- A. $O(m^3)$ B. $O(n^2)$ C. $O(m * n)$ D. $O(m + n)$

【解答】C

10. 执行下面程序段时,s 语句的执行次数为()。

```
for(int i = 1; i <= n - 1; i++)
  for(int j = i + 1; j <= n; j++)
    s;
```

- A. $n(n-1)/2$ B. $n^2/2$ C. $n(n+1)/2$ D. n



【解答】A

二、简答题

1. 举一个数据结构的例子,叙述其逻辑结构、存储结构和运算三个方面的内容。

【解答】 例如有一份通讯录,如表 1.1 所示,记录了相关人员的电话号码,将其按姓名一人占一行构成表,这个表就是一个数据结构。每一行为一个记录,每个记录(包括姓名、工作单位、职务、电话号码)即为一个结点,对于整个表来说,只有一个开始结点(前面无记录)和一个终端结点(后面无记录),其他的结点则各有一个也只有一个直接前趋和直接后继(它的前面和后面均有且只有一个记录)。这几个关系就确定了这个表的逻辑结构。

表 1.1 通讯录

姓 名	工作单位	职 务	电话号码
吴 浩	市交警大队	副大队长	88234567
汪 涌	市自来水公司	职 工	87342567
.....

那么怎样把这个表中的数据存储到计算机里呢?用高级语言如何表示各结点之间的关系呢?是用一段连续的内存单元来存放这些记录(如用数组表示),还是随机存放各结点数据再用指针进行链接呢?这就是存储结构的问题。我们都是从高级语言的层次来讨论这个问题的。例如,若用链式存储方式,结点的数据类型定义如下:

```
struct node{
    char name[8];           // 存放姓名的数据域
    char dept[20];         // 存放工作单位的数据域
    char duty[10];         // 存放职务的数据域
    int num;                // 存放电话号码的数据域
    struct node * next;    // 指向下一个结点的指针
}
```

2. 简述顺序存储结构与链式存储结构在表示数据元素之间关系上的主要区别。

【解答】 在顺序存储结构中,逻辑关系上相邻的两个元素在物理位置上也相邻,可以随机存取表中任一元素;而链式存储结构中,数据元素之间的关系是由结点中的指针指示的。

3. 设有数据逻辑结构为

$$B = (K, R)$$

$$K = \{k_1, k_2, \dots, k_7\}$$

$$R = \{ \langle k_1, k_2 \rangle, \langle k_1, k_3 \rangle, \langle k_1, k_6 \rangle, \langle k_2, k_4 \rangle, \langle k_3, k_5 \rangle, \langle k_4, k_5 \rangle, \langle k_5, k_6 \rangle, \langle k_5, k_7 \rangle, \langle k_6, k_7 \rangle \}$$

画出这个逻辑结构的图示,并确定相对于关系 R ,哪些结点是开始结点?哪些结点是终端结点?

【解答】 该题的逻辑结构图示如图 1.6 所示。

开始结点是指无前趋的结点,这里满足该定义的开始结点为 k_1 。

终端结点是指无后继的结点,这里满足该定义的终端结点为 k_7 。



该逻辑结构是图形结构。

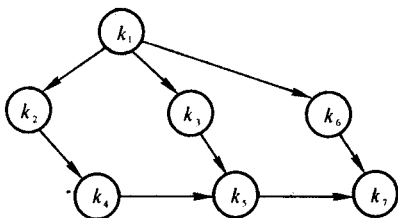


图 1.6 一个逻辑结构示意图

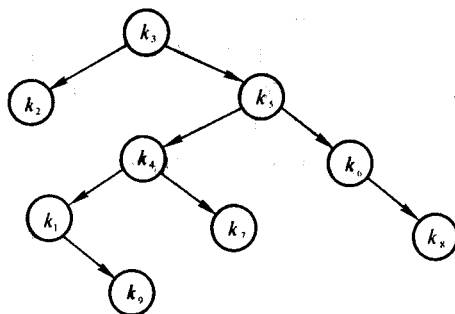


图 1.7 一个逻辑结构示意图

4. 设有如图 1.7 所示的逻辑图, 给出它的逻辑结构。

【解答】本题的逻辑结构如下:

$$B = (K, R)$$

$$K = \{k_1, k_2, \dots, k_8\}$$

$$R = \{ \langle k_3, k_2 \rangle, \langle k_3, k_5 \rangle, \langle k_5, k_4 \rangle, \langle k_5, k_6 \rangle, \langle k_4, k_1 \rangle, \langle k_4, k_7 \rangle, \langle k_1, k_0 \rangle, \langle k_6, k_8 \rangle \}$$

该逻辑结构是一个树形结构, 其树根为 k_3 , 叶子结点为 k_2, k_7, k_8 和 k_0 。

5. 有下列几种用二元组表示的数据结构, 画出它们分别对应的逻辑图形表示, 并指出它们分别属于何种结构。

(1) $A = (K, R)$, 其中:

$$K = \{a, b, c, d, e, f, g, h\}$$

$$R = \{r\}$$

$$r = \{ \langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle, \langle d, e \rangle, \langle e, f \rangle, \langle f, g \rangle, \langle g, h \rangle \}$$

(2) $B = (K, R)$, 其中:

$$K = \{a, b, c, d, e, f, g, h\}$$

$$R = \{r\}$$

$$r = \{ \langle d, b \rangle, \langle d, g \rangle, \langle d, a \rangle, \langle b, c \rangle, \langle g, e \rangle, \langle g, h \rangle, \langle e, f \rangle \}$$

(3) $C = (K, R)$, 其中:

$$K = \{1, 2, 3, 4, 5, 6\}$$

$$R = \{r\}$$

$$r = \{ (1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6) \}$$

这里的圆括号对表示两结点是双向的。

(4) $D = (K, R)$, 其中:

$$K = \{48, 25, 64, 57, 82, 36, 75\}$$