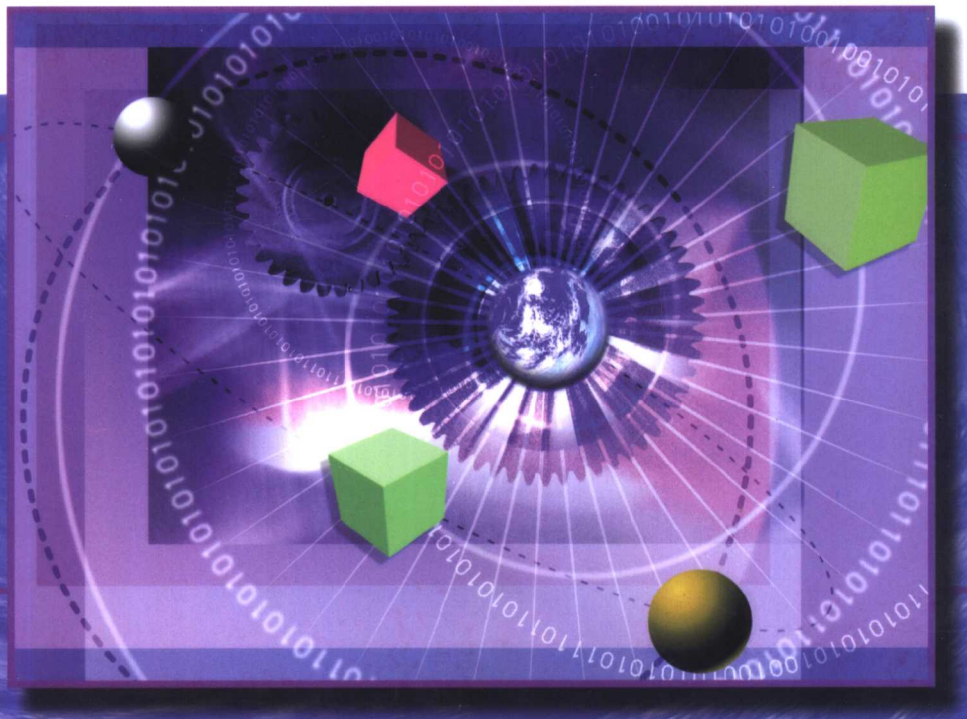
计算机专业基础课程辅导丛书

《数据结构》学习指导与训练



蒋盛益 主编

蒋盛益 徐雨明 周志方 宋毅军 王 樱 编著

中国水利水电出版社
www.waterpub.com.cn

计算机专业基础课程辅导丛书

《数据结构》学习指导与训练

蒋盛益　主　编

蒋盛益　徐雨明　周志方　宋毅军　王樱　编著

中国水利水电出版社

内 容 提 要

全书以数据结构课程考研的一般要求为依据，以知识要点为线索，按照知识要点复习、典型例题剖析、习题及参考答案三大模块组织各章内容。典型例题与习题的题型与一般院校的考研题型相一致。本书可供考研者复习参考，亦可作为初学数据结构课程的辅助材料。

图书在版编目（CIP）数据

《数据结构》学习指导与训练/蒋盛益主编.－北京：中国水利水电出版社，2003

（计算机专业基础课程辅导丛书）

ISBN 7-5084-1640-6

Ⅰ.数… Ⅱ.蒋… Ⅲ.数据结构-自学参考资料 Ⅳ.TP311.12

中国版本图书馆 CIP 数据核字（2003）第 067971 号

书　　名	《数据结构》学习指导与训练
作　　者	蒋盛益　主编　蒋盛益　徐雨明　周志方　宋毅军　王樱　编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787×1092 毫米　16 开本　24.75 印张　568 千字
版　　次	2003 年 8 月第一版　2003 年 8 月北京第一次印刷
印　　数	0001—5000 册
定　　价	34.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

丛书序

本丛书包含计算机专业的主干基础课程：《数据结构》、《操作系统》、《汇编语言与微机原理》、《计算机组成原理》等，这些课程同时也是信息技术相关专业，如信息与计算科学、信息管理、电子技术、通信等的重要基础课程，同时这些课程也是许多专业研究生入学考试的课程。随着信息技术的应用越来越广泛，学习这些课程的读者越来越多。本丛书的编写目的在于提高考研复习和课程学习的质量，巩固和深化读者应用知识的能力。本丛书是在作者多年教学过程中建立试题库的基础上加以整理、扩充而成的，丛书中的大部分题目来自自学考试试题和部分高校、科研机构历届考研试题。

本丛书各分册以课程考研的一般要求为依据，以知识要点为线索，按照知识要点复习、典型例题剖析、习题及参考答案三大模块进行组织。本丛书具有内容阐述简洁而全面，重点突出解题思路，强调内容、方法的综合性，方便读者使用等特点。

内容阐述简洁而全面。本丛书不同于一般教材，不过多地解释简单的术语，而是对课程的概念和方法进行高度概括和总结，将课程的重点、难点充分地融入到典型例题之中，通过例题的剖析对知识和解决问题的方法进行了扩充与深化。使读者将主要的精力集中在知识的运用、解题过程中，使读者得以全面温习与提高，花较少的时间复习各门课程的内容。

重点突出解题思路。本丛书重点介绍解题的方法，对于典型例题和习题按知识点进行归纳组织，且同一题型题目基本上按从易到难的顺序编排，这样便于读者使用，提高学习效率。由于许多题目选自研究生入学考试试题，因而实用性较强。

强调内容、方法的综合性。本丛书所选的例题和习题许多具有较高的综合性，一个问题或者融合了多个概念，或者可以采用多种方法解决，或者一种方法可以用来解决不同的问题，通过对这些问题的学习和理解使读者能做到触类旁通、举一反三，希望借此提高读者解决问题的能力。

本丛书各分册可供考研者复习相应课程参考之用，亦可作为初学相应课程的辅助材料。

感谢中国水利水电出版社的大力支持，特别感谢孙春亮总编辑的关爱，使本丛书能得以与读者见面。

编者

2003 年 5 月

前　言

全书以课程知识要点为线索，按照知识要点复习、典型例题剖析、练习题及参考答案三大模块组织。

知识要点复习：对每一章的主要内容进行了归纳总结，指出了每一章的知识点、重点和难点，便于读者整体上把握课程知识框架。

典型例题剖析：通过对典型问题剖析解答，融每章的重点、难点和常用方法于典型例题之中。

练习题及参考答案：习题按照知识层次组织为选择题、填空题、判断题、简答题、算法设计题，为便于读者学习使用，方便查找，每一题型中习题基本上按照知识点的顺序从易到难进行了归纳组织。

学习指导与训练丛书是在作者多年教学过程中建立试题库的基础上加以整理形成的。本书凝聚了许多教材、参考书的精华及作者多年的教学经验，大部分题目来自自学考试试题和部分高校、科研机构历届考研试题，这里不一一列出，在此作者表示感谢。

本丛书的特色在于：对试题按知识点进行了归纳组织，使读者能作到触类旁通、举一反三，同一题型题目按从易到难的顺序编排，便于读者使用，提高学习效率。

本丛书适合于不同层次的学习者参考使用，既可作为计算机及相关专业学生学习操作系统课程的辅助材料，也可供考研者参考。

《数据结构》学习指导与训练书中的知识要点主要参考清华大学出版社出版、严蔚敏，吴伟民编著的《数据结构(C 语言版)》一书，在学习使用过程中需注意部分概念在不同教材体系中描述的差异。

本书由蒋盛益、徐雨明、周志方、宋毅军、王樱编写。其中第 1 章、第 9 章由徐雨明编写，第 2 章、第 3 章由宋毅军编写，第 4 章、第 5 章、第 8 章、第 10 章、第 11 章、第 12 章由蒋盛益编写，第 6 章由周志方编写，第 7 章由王樱编写，最后由蒋盛益、徐雨明修改统稿。

由于编者水平有限，书中难免存在错误和不妥之处，敬请读者提出宝贵意见。

编者

2003 年 5 月

第 1 章　绪论

1.1　知识要点复习

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操作对象以及它们之间的关系和操作等的学科。

1.1.1　基本概念

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。随着计算机科学的发展，数据概念的内涵在逐步扩大。

2. 数据元素

数据元素是数据的基本单位，通常作为一个整体进行考虑和处理。一个数据元素由若干个数据项组成，数据项是数据的不可分割的最小单位。

3. 数据对象

数据对象是性质相同的数据元素的集合，是一个数据的子集。

4. 数据结构

数据的逻辑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，形式定义为一个二元组：

```
Data_Structure=(D,S)
```

其中，D 是数据元素的有限集；S 是 D 上关系的有限集。

数据的逻辑结构是从逻辑关系上来观察数据，它与数据的存储方式无关，是独立于计算机的。根据数据元素之间关系的不同特性，基本逻辑结构通常有 4 种：

（1）集合：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：数据元素之间存在一个一对一的关系，有且仅有一个开始结点和终端结点，除开始结点外，每个结点有且仅有一个前趋结点，除终端结点外，每个结点有且仅有一个后继结点。

（3）树型结构：数据元素之间存在一个对多个的关系，有一个开始结点和多个终端结点，除开始结点外，每个结点有且仅有一个前趋结点，除终端结点外，每个结点可能有多个后继结点。

（4）网状结构或图状结构：数据元素之间存在多个对多个的关系，每个结点可能有多个前趋结点和多个后继结点。集合是数据元素之间关系极为松散的一种结构，因此也可以用其他结构来表示。树型结构和网状结构又称为非线性结构。

数据的存储结构或物理结构是逻辑结构在计算机存储器里的实现，它是依赖于计算机的。元素之间的逻辑关系在计算机中有两类不同的表示方法：顺序映象和非顺序映象。详细的存储方法可分为：顺序存储、链接存储、索引存储、散列存储。

顺序存储是把结点存储在物理上相邻的一组存储单元里，结点之间的关系由存储单元的邻接关系来体现。

链接存储是将结点所占的存储单元分为两部分：一部分存放结点本身的信息，即数据项；另一部分存放该结点的后继结点所对应的存储单元的地址，即指针项。

索引存储是用结点的索引号来确定结点存储地址。

散列存储是根据结点的关键字确定它的存储地址。

除顺序存储外,其他存储方式除存储数据本身外还要占用地址信息（空间），因此，空间利用率降低。在以后各章中可以看到,这是为了以一定的空间开销来换取操作的方便(因而提高效率）。

结构的存储密度＝数据本身所占的存储容量/整个结构所占的存储容量。

如果所有的存储空间都分配给了数据，则这个存储结构叫紧凑结构。紧凑结构的存储密度为 1，非紧凑结构的存储密度小于 1。顺序存储的线性表是紧凑结构，而链式存储结构是非紧凑结构。

数据的逻辑结构和存储结构是密切相关的，任何一个算法的设计取决于选定的数据（逻辑）结构，而算法的实现取决于采用的存储结构。因此，可以说数据结构由数据的逻辑结构、数据的存储结构和数据的运算三部分组成，或者说按某种逻辑关系组织起来的一批数据，按一定的存储表示方式把它存储在计算机的存储器中，并在这些数据上定义运算的集合，就叫做数据结构。

5. 抽象数据类型

抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关，即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。

为了提高软件的复用率，近代程序设计方法学指出，一个软件系统的框架应建立在数据之上，而不是建立在操作之上。所定义的数据类型抽象层次越高，含有该抽象数据类型的软件模块的复用程度就越高。

一个抽象数据类型的软件模块通常应包含定义、表示和实现 3 个部分。抽象数据结构 ADT 可形式定义为一个三元组：

```
Abstruct_Data_Structure=(D,S,P)
```

其中，D 是数据元素的有限集；S 是 D 上关系的有限集；P 是施加在 D 上的基本操作的有限集。

1.1.2 算法及其分析

1. 算法及其特性

算法是对特定问题求解步骤的一种描述，是指令的有序序列。由于数据的逻辑结构和

存储结构不是惟一的，在很大程度上可以由用户自行选择和设计，所以处理同一问题的算法也不是惟一的。即使对相同的逻辑结构和存储结构而言，其算法的设计思想和技巧不同，编写出的算法也大不相同。

算法的 5 个重要特性：（1）有穷性；（2）确定性；（3）输入；（4）输出；（5）可行性。

2. 算法的评价

评价一个算法一般从以下四个方面进行：

（1）正确性；（2）可读性；（3）健壮性；（4）高效率与低存储量。

在计算机科学中时间效率（运算量少）与空间效率(低存储量)往往是矛盾的，在实际应用中需要根据问题的要求和特点而选择侧重点。如：索引技术、缓冲技术和散列技术就是以空间换时间的技术；而数据压缩技术、覆盖技术则是以时间换空间的技术。

3. 算法效率的度量

一个特定算法“运行工作量”的大小，只依赖于问题的规模，或者说，是问题规模的函数。为便于比较同一问题的不同算法，通常从算法中选取一种对于所研究的问题（或算法类型）来说是基本的原操作，以该基本操作重复执行的次数作为算法的时间量度。算法中基本操作重复执行的次数是问题规模 n 的某个函数 f(n)，算法的时间量度记作

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

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，称作算法的渐近时间复杂度，简称时间复杂度。

由于算法的时间复杂度考虑的只是对于问题规模 n 的增长率，在难以精确计算基本操作执行次数（或语句频度）的情况下，只需求出它关于 n 的增长率或阶即可。通常算法中基本操作重复执行的次数随问题的输入数据集不同而不同，对这类算法的分析，其一是计算算法的平均时间复杂度，其二是计算最坏情况下的时间复杂度。常用的时间复杂度有如下关系：

$O(1)\leqslant O(\log_2 n)\leqslant O(n)\leqslant O(n\log_2 n)\leqslant O(n^2)\leqslant O(2^n)$

在设计算法时一般情况下应尽可能避免使用时间复杂度是指数阶的，而使用复杂度是低阶多项式的，以便能在有效的时间内解决问题；但在计算机安全的加密算法中，却需要有足够的复杂性，以保证不能使入侵者在有效时间内攻入系统。

1.1.3　要点提示

【本章考点】有关数据结构、存储结构的定义及其关系，语句执行频度、算法时间复杂度、空间复杂度的计算。

【本章重点和难点】语句执行频度、算法时间复杂度的计算。

1.2　典型例题剖析

【例 1】设有以下 3 个函数：$f(n)=21n^4+n^2+1000$、$g(n)=15n^4+500n^3$、$h(n)=5000n^{3.5}+n\log_2 n$，

请判断以下断言正确与否：

（1）f(n)是 O(g(n))

（2）h(n)是 O(g(n))

（3）h(n)是 $O(n^{3.5})$

（4）h(n)是 $O(n\log_2 n)$

【解答】注意到，对于多项式函数，其增长速度是由最高次幂决定的，而 $n\log_2 n<n^2$，可以看出：（1）对，（2）错，（3）对，（4）错

【例 2】设 n 为正整数。试确定下列各程序段中前置以记号@的语句的频度。

（1）
```
    i=1;k=0;
    do
@   {   k+=10*i;
        i++;
    }while(i<=n-1);
```

（2）
```
    x=0;
    for(i=1;i<=n;i++)
    {   for(j=i;j<=i;j++)
        {   for(k=1;k<=j;k++)
@           x+=delta;
        }
    }
```

（3）
```
    x=91; y=100;
    while(y>0)
@   {   if(x>100) {x-=10;y--;}
        else  x++;
    }
```

（4）
```
    i=1;
    while(i<=n)
@    i=i*3;
```

【解答】

（1）这是一个单重循环，循环次数为 n−1，因此语句执行的频度为 n−1。

（2）这是一个三重循环。对固定的 i 及固定 j，语句执行 j 次。

j 从 1 到 i 时语句执行了：

$$\sum_{j=1}^{i} j=\frac{i(i+1)}{2}$$

i 从 1 到 n 时语句执行了：

$$\sum_{i=1}^{n}\frac{i(i+1)}{2}=\frac{n(n+1)(n+2)}{6}$$

因此语句执行的频度为：

$$\frac{n(n+1)(n+2)}{6}=O(n^3)$$

（3）此程序段实质上是一个双重循环，对于固定的 y（>0），if 语句执行 11 次（前 10 次 x 由 91～101，最后一次 x 由 101～91，y 减 1），y 由 100 变到 0，循环执行 100 次，故总的执行次数为 11×100＝1100。

（4）i 依次取 1，3，32，…，3i−1，故根据循环条件有 $3^{i-1} \leqslant n$，$i \leqslant \log_3 n+1$，语句执行频度为：

$\lfloor \log 3n \rfloor + 1 = O(\log 3n)$

【例 3】指出下面算法的时间复杂度。

```
void prime(int n )
{   int i=2;
    while((n%i) && i<=sqrt(n))
            i++;
    if(i>sqrt(n))  printf("%d 是一素数\n",n);
    else    printf("%d 不是一素数\n",n);
}
```

【解答】易见该算法的时间复杂度主要由 while 循环决定，若 n 为偶数，while 指令只执行一次，若 n 为素数，则 while 指令执行的次数达到最大值 $\sqrt{n}-1$，因此该算法最好情况的时间复杂度为 $O(1)$，最坏情况的时间复杂度为 $O(\sqrt{n})$，平均时间复杂度为 $O(\sqrt{n})$。

【例 4】求两个 n 阶矩阵的乘法 C=A×B 的算法如下：

```
#define  MAX  100
void  maxtrixmult(int n,float a[MAX][MAX], B[MAX][MAX], float c[MAX][MAX])
{
   int i,j,k;
   float x;
   for(i=1;i<=n;i++)                          (1)
   {
         for(j=1;j<=n;j++)                    (2)
         {
               x=0;                           (3)
               for(k=1;k<=n;k++)                  (4)
                   x+=a[i][k]*b[k][j];            (5)
               c[i][j]=x;                         (6)
            }
          }
}
```

写出算法中带标号语句的频度，并求该算法的时间复杂度。

【解答】该算法主体是一个三重循环，这里以标号（2）的语句说明其频度：

对于固定的 i、j 从 1 变到超过 n，指令 j++需执行 n+1 步，而相对于外循环（1），i 从 1 变到超过 n，内循环重复 n 次，故（2）的执行频度为 n(n＋1)。

同理可以计算带标号语句的频度分别为：

（1）n+1，（2）n(n+1)，（3）n^2，（4）$n^2(n+1)$，（5）n^3，（6）n^2

算法时间复杂度为所有语句的频度之和。

$T(n)=2n^3+4n^2+2n+1=O(n^3)$

【例 5】已知有实现同等功能的两个算法，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 10^7 秒（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？请说明理由。

【解答】总运算量为 $10^7\times10^5=10^{12}$，根据题意：

对第一个算法有：$2^n\leqslant10^{12}$，$n\leqslant12*\log_210\approx39.86$

对第二个算法有：$n^{10}\leqslant10^{12}$，$n\leqslant10^{1.2}\approx15.84$

因此第一个算法的可解规模大一些。

由此可见，虽然一般情况下多项式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。

【例 6】设 A[K]是一个整数数组，各分量的初值为 0，即 A[i]=0（$0\leqslant i\leqslant k-1$），设 n 为连续调用 demo 过程的次数，请回答下述问题：

（1）给出调用次数 n=4 时，数组 A 中各分量的结果。

（2）demo 过程的功能是什么？数组 A 起什么作用？

（3）若 $n=2^k$，数组 A 中各分量的结果是什么？

（4）显示过程的最坏时间是 O(k)，故连续调用 demo 过程 n 次的总时间为 O(kn)，但是这个界偏大，请详细分析求出一个更小的 n 次连续调用 demo 的总时间的上界。

```
void  demo(arraytype a;int k)
{   i=0;
    while (i<k&&a[i])
    {  a[i]=0;  i++;   }
    if (i<k)   a[i]=1;
}
```

【解答】对于前几次调用有如下结果：

调用次数	数组 A 的元素
1	A[0]=1，其余元素为 0。
2	A[0]=0，A[1]=1，其余元素为 0。
3	A[0]=1，A[1]=1，其余元素为 0。
4	A[0]=0，A[1]=0，A[2]=1，其余元素为 0。
5	A[0]=1，A[1]=0，A[2]=1，其余元素为 0。
6	A[0]=0，A[1]=1，A[2]=1，其余元素为 0。
7	A[0]=1，A[1]=1，A[2]=1，其余元素为 0。
8	A[0]=0，A[1]=0，A[2]=0，A[3]=1，其余元素为 0。

（1）由以上可见，调用次数 n=4 时除 A[2]=1 外，其余各分量均为 0。

（2）通过分析可以看出，demo 的功能是二进制的增量操作，A 相当于一个二进制计数器，用以保存 n 的二进制表示。

（3）当 $n=2^k$ 时，n 的二进制表示为$(10\cdots0)_2$，最低 k 位全为 0，因此 A 中分量均为 0。

（4）n 次连续调用 demo 的总时间的上界应为 O(n)。

demo 操作的时间代价正比于二进制计数器 A 中位（每个 A（i）相当于二进制数的位）的翻转次数，而 n 次连续的 demo 操作是从 0 开始的，因此在这 n 次增量操作中：A[0]共

翻转 n 次，A[1]共翻转$\frac{n}{2}$次，…，A[i]共翻转$\frac{n}{2^i}$次（$i \leqslant \log_2 n$）；当 i>$\log_2$n 时，位 A[i]根本不翻转（因为数 n 的二进制表示最多有 $\log_2$n+1 位），所以 n 次增量操作发生的位翻转总数为：

$$\sum_{i=0}^{\log_2 n} \frac{n}{2^i} < n\sum_{i=0}^{\infty} \frac{1}{2^i} = 2n = O(n)$$

【例 7】某算法所需时间由下述方程表示，试求出该算法的时间复杂度的级别（以大 O 形式表示）。注意 n 为求解问题的规模，为简单起见，设 n 为 2 的正整数幂。

$$T(n)=\begin{cases}1 & n=1\\ 2T(n/2)+n & n>1\end{cases}$$

【解答】设 $n=2^k$（$k \geqslant 0$），则有关系式：

$$T(2^k) = 2T(2^{k-1}) + 2^k == 2^2 T(2^{k-2}) + 2*2^k$$

经过推导可以得出一般递推关系式：

$$T(2^k) = 2^i T(2^{k-i}) + i2^k$$

由此可推得：

$$T(2^k) = 2^k T(2^0) + k2^k = (k+1)2^k$$

即：

$$T(n) = n(\log_2 n + 1) = n\log_2(2n) = O(n\log_2 n)$$

当 n 不是 2 的正整数幂时，上述 $T(n)$仍满足方程。

【例 8】有以下程序，分析其中 order()函数的时间复杂度。

```
int  a[ ]={2,5,1,7,9,3,6,8};
int n=8;
order(int j,int n)
{  int i,temp;
   if (j<m)
   {   for(i=j;i<n;i++)
         if(a[i]<a[j])
         {  temp=a[i];
            a[i]=a[j];
            a[j]=temp;
          }
         j++;
         order(j,n);
    }
}

main()
{  int i;
   order(0,n);
   for(i=0;i<n;i++)
     printf("%d",a[i];
}
```

【解答】order()函数的功能是递归实现排序,设 T(n)是时间复杂度，在排序 n 个元素时，算法的计算时间主要花费在递归调用 order()上。第一次调用时，处理过程分两大步：第一步是将序列中每个元素与最前面的元素比较，若小于最前面的元素则交换位置，这需要 n-1 次比较，这一步后最小的元素位于最前面；第二步是对余下的 n-1 个元素排序，其时间复杂度为 T(n-1)。故得到如下方程：

$$T(n)=\begin{cases}0 & n=1\\ T(n-1)+n-1 & n>1\end{cases}$$

对此方程递推求解得：

$$\begin{aligned}T(n)&=T(n-1)+n-1=(T(n-2)+n-2)+n-1\\&=T(n-2)+(n-2)+(n-1)\\&=T(n-3)+(n-3)+(n-2)+(n-1)\\&\cdots\\&=(T(1)+1)+2+3+\cdots+(n-2)+(n-1)\\&=0+1+2+\cdots+(n-1)\\&=n(n-1)/2\approx O(n^2)\end{aligned}$$

故 order()函数的时间复杂度为 $O(n^2)$。

1.3 练习题及参考答案

1.3.1 练习题

一、选择题

1. 每种数据结构都具备 3 个基本运算：插入、删除和查找，这种说法（　　）。

A. 正确　　B. 不正确

2. 数据结构被形式定义为（D，S），其中 D 是（　　）的有限集合，S 是 D 上的（　　）有限的集合。

A. 算法　　B. 数据元素　　C. 数据操作　　D. 逻辑结构

E. 操作　　F. 映象　　G. 存储　　H. 关系

3. 以下与数据的存储结构无关的术语是（　　）。

A. 循环队列　　B. 链表　　C. 哈希（hash）表　　D. 栈

4. 算法分析的目的是（　　），算法分析的两个主要方面是（　　）。

A. 给出数据结构的合理性　　B. 研究算法中的输入和输出的关系

C. 分析算法的效率以求改进　　D. 分析算法的易懂性和文档性

E. 空间复杂性和时间复杂性　　F. 正确性和简明性

G. 可读性和文档性　　H. 数据复杂性和程序复杂性

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

6. 计算机算法指的是（　　），它必具备输入、输出和（　　）5 个特性。

A. 计算方法　　B. 排序方法

C. 解决问题的有限运算序列　　D. 调度方法

E. 可行性、可移植性和可扩充性　　F. 可行性、确定性和有穷性

G. 确定性、有穷性和稳定性　　H. 易读性、稳定性和安全性

7. 线性表的顺序存储结构是一种（　　）的存储结构,线性表的链式存储结构是一种（　　）存储结构。

A. 随机存取　　B. 顺序存取

C. 索引存取　　D. 散列存取

8. 线性表若采用链式存储结构时,要求内存可用存储单元的地址（　　）。

A. 必须是连续的　　B. 部分地址必须是连续的

C. 一定是不连续的　　D. 连续不连续都可以

9. 算法的时间复杂度取决于（　　）。

A. 问题的规模　　B. 待处理数据的初态

C. 问题的规模和待处理数据的初态

10. 下述函数中渐近时间最小的是（　　）。

A. $T_1(n)=n\log_2 n+1000\log_2 n$　　B. $T_2(n)=n\log_2 3-1000\log_2 n$

C. $T_3(n)=n^2-1000\log_2 n$　　D. $T_4(n)=2n\log_2 n-1000\log_2 n$

二、填空题

1. 数据逻辑结构包括_____三种类型，树型结构和图形结构合称_____。

2. 对于给定的 n 个元素，可以构造出的逻辑结构有_____、_____、_____和_____四种。

3. 算法的五个重要特性是_____。

4. 评价算法的性能从利用计算机资源角度看主要从_____方面进行分析。

5. 线性结构中元素之间存在_____关系，树型结构中元素之间存在_____关系，图形结构中元素之间存在_____关系。

6. 下面程序段的时间复杂度是_____。

```
i=s=0;
while(s<n)
{
   i++;    /*i=i+1*/
   s++;    /*s=s+1*/
}
```

7. 下面程序段的时间复杂度是_____。

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

三、计算题

1．试比较两函数 n^2 和 $50n\log_2 n$ 的增长趋势，并确定 n 在什么范围内，函数 n^2 的值大于 $50n\log_2 n$ 的值。

2．按增长率由小至大的顺序排列下列各函数：

2^{100}，$(3/2)^n$，$(2/3)^n$，n^n，$n^{3/2}$，$\sqrt{n}$，$n^{2/3}$，$n!$，n，$\log_2 n$，$n/\log_2 n$，$(\log_2 n)^2$，$\log_2(\log_2 n)$，$n\log_2 n$，$n^{\log 2n}$

3．设 n 为正整数。试确定下列各程序段中前置以记号@的语句的频度。

(1)
```
        i=1;k=0;
        while(i<n-1)
@       {   k+=10*i;
            i++;
        }
```

(2)
```
        k=0;
        for(i=1;i<=n;i++)
        {   for(j=i;j<=n;j++)
@           k++;
        }
```

(3)
```
        i=1;  j=0;
        while(i+j<=n)
@       {   if(i>j)   j++;
            else    i++;
        }
```

4．写出下面算法中带标号语句的频度。

```
void perm(int a[],k,n)
{   int x,i;
(1)  if (k==n)
(2)    for(i=1;i<=n;i++)
(3)        printf("%d ",a[i]);
     else
   {  (4)  for(i=k;i<=n;i++)
      (5)     a[i]+=i*i;
      (6)  perm(a,k+1,n);
   }
}
```

执行函数调用语句：perm(a,1,n)

5．指出下列两个算法的时间复杂度。

(1)
```
int sum1(int n)
        {
          int p=1, sum=0, i;
          for(i=1;i<=n;i++)
             {
               p*=i;
               sum+=p;
```

```
        }
        return(sum);
    }
```

(2)
```
int sum2(int n)
{
  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);
}
```

6. 有如下递归函数 fact(n)，分析其时间复杂度。

```
fact(int  n)
{
  if(n<=1)   return(1);            (1)
  else   return(n*fact(n-1);       (2)
}
```

7. 假设 n 为 2 的乘幂，并且 n>2，试求下列算法的时间复杂度及变量 count 的值（以 n 的函数形式表示）。

```
int Time(int n)
{  int count=0,x=2;
   while(x<n/2)
   {  x*=2;  count++;  }
   return (count);
}
```

8. 有下列几种用二元组表示的数据结构，画出它们对应的逻辑图形表示，并指出它属于哪种结构。

（1）A=(K,R)，其中：

```
K={a,b,c,d,e,f,g,h}
R={r}
r={<a,b>,<b,c>,<c,d>,<d,e>,<e,f>,<f,g>,<g,h>}
```

（2）B=(K,R)，其中：

```
K={a,b,c,d,e,f,g,h}
R={r}
r={<d,b>,<d,g>,<d,a>,<b,c>,<g,e>,<g,h>,<e,f>}
```

（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)}
```

这里圆括号表示结点关系是双向的。

（4）D=(K,R)，其中：

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

R={r1,r2}

r1={<25,36>,<36,48>,<48,57>,<57,64>,<64,75>,<75,82>}

r2={<48,25>,<48,64>,<64,57>,<64,82>,<25,36>,<84,75>}

9. 设有数据逻辑结构为：

B=(K,R)

K=(k1,k2,k3,…,k9)

R={<k1,k3>,<k1,k8>,<k2,k3>,<k2,k4>,<k2,k5>,<k3,k9>,
<k5,k6>,<k8,k9>,<k9,k7>,<k4,k7>,<k4,k6>}

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？

10. 设有如图 1-1 所示的逻辑结构图，给出它的逻辑结构。

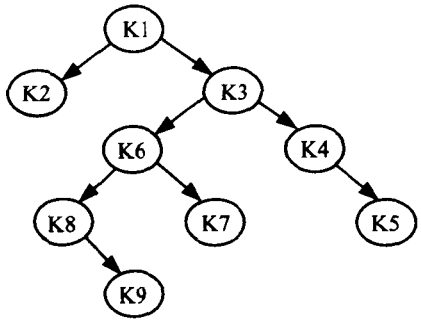


图 1-1 逻辑结构图

11. 试描述数据结构和抽象数据类型的概念与程序设计语言中数据类型概念的区别。

12. 斐波那契（Fibonacci）数列 F_n 定义如下：

$$F_n=\begin{cases}0 & n=0\\ 1 & n=1\\ F_{n-1}+F_{n-2} & n\geqslant 2\end{cases}$$

回答下列问题：

（1）在递归计算 F_n 的时候，需要对较小的 F_{n-1}，F_{n-2}，…F_1，F_0 精确计算多少次？

（2）用大 O 表示法，试给出递归计算 F_n 的时间复杂度是多少？

四、算法设计题

1. 试编写算法，计算 $i!\cdot 2^i$ 的值并存入数组 a[0…arrsize-1]的第 i-1 个分量中（i=1,2,…,n）。假设计算机中允许的整数最大值为 max int，则当 n>arrsize 或对某个 k（1≤k≤n）使 $k!\cdot 2^k$>max int 时，应按出错处理。注意选择认为较好的出错处理方法。

2. 假设有 A、B、C、D、E 五个高等院校进行田径对抗赛，各院校的单项成绩均已存入计算机，并构成一张表，表中每一行的形式为：

项目名称	性 别	校 名	成 绩	得 分

编写算法，处理上述表格，以统计各院校的男、女总分和团体总分，并输出。