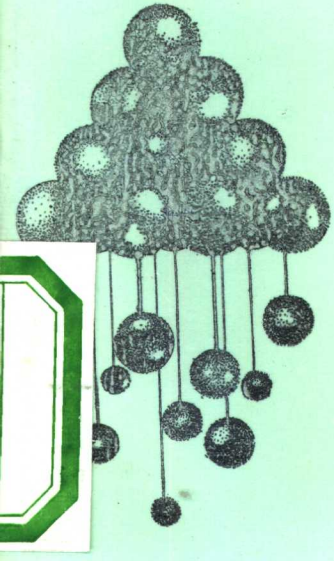
A COLLECTION OF EXCERCISES IN DATA STRUCTURES

数据结构题集

严蔚敏 米 宁 吴伟民 编



清华大学出版社

数据结构题集

严蔚敏　米　宁　吴伟民　编

清华大学出版社

内 容 提 要

这是一本和清华大学出版社出版的教科书《数据结构》相配套的习题集，主要内容有：十二组(约300个)习题、五组(23个)实习题、部分习题的答案和提示，最后有四个附录。每一组习题对应教科书中的一章，结合该章的学习要求进行编排；每一组实习题都有鲜明的主题，围绕1至2种数据结构安排3至6题，每个题都有具体的要求，并明确区分必做内容和选做内容。在附录中除给出算法书写规范、实习步骤规范外，还给出了一个实习报告范例。

本书内容丰富、程序设计观点新颖，在内容的详尽程度上接近于课程辅导材料，不仅可作为大专院校的配套教材，也是广大工程技术人员和自学读者的颇有帮助的辅助教材

（京）新登字158号

数 据 结 构 题 集

严蔚敏 米 宁 吴伟民 编

责任编辑 贾仲良

☆

清华大学出版社出版

北京 清华园

北京市昌平印刷厂 印装

新华书店总店科技发行所发行

☆

开本：850×1163 1/32 印张：6.25 字数：168千字

1988年12月第1版 1993年8月第4次印刷

印数：63001—78000

ISBN 7-302-00281-9/TP·105

定价：3.90元

前　言

数据结构是计算机科学专业的主要基础课程之一。它旨在使学生了解数据对象的特性，学会数据组织的方法和把现实世界中的问题在计算机内部的表示方法，以及培养基本的、良好的程序设计技能。其中，技能培养的重要程度决不亚于知识传授。在整个教学过程中，习题和实习是两个至关重要的环节。为了帮助读者学好这门课程，我们编写了这本指导性题集。

这本题集与《数据结构》(清华大学出版社，1987 年出版)一书是配套教材，习题和实习都是按这本书的内容顺序来组织的，很多习题涉及教科书上的内容或算法，因此最好读者手边能有这本教科书以便随时参阅。

习题的作用在于帮助学生深入理解教材内容，巩固基本概念，达到培养良好的程序设计的习惯，并掌握其技能，同时也是检查对授课内容理解和掌握程度的手段。题集的第一部分含有全部 300 个习题，组织成十二组，每组对应教科书中的一章，在每组习题之前给出了这一章的内容提要和学习要求。这些习题是在编者多年教学过程中所积累资料的基础上，参考近年来大量国外教材之后精心设计而成的。书中对特别推荐的题作了标记，并对每道习题的难度按五级划分法给出了难度系数。虽然这些都带有强烈的主观色彩，编者还是希望能对任课教师和自学的读者选择习题起到辅助作用。

涉及算法的习题侧重于局部程序设计，即如何编好"小程序"(Programming-in-the-small)。但仅有这方面的训练还是不够的。在本书的第二部分，以线性表、栈和队列、串、树和图等部分内容为核心设置了五组实习题，每组有 3 至 6 题，供读者自

由选择。对于这些题都区分了基本要求和选作的进一步要求。这样不但使题目能适合于不同程度的读者，而且能使读者在系统设计时就考虑到程序的可扩充性，尽可能寻求有普遍意义的解法，避免“就事论事”。题目本身和要求都十分详细具体，以避免产生歧意。编者希望这些实习题能对习题起到良好的补充作用，使读者受到涉及程序系统设计的完整过程的综合训练，体会系统结构设计（Programming-in-the-large）与“小程序”设计的区别，培养合作能力，为将来进行软件开发和研究工作作一次“实践演习”。

数据结构课程教学中的一种常见现象是：理解授课内容并不困难，但一接触习题，往往不是感到无从下手，就是解答中出错很多，有时即使正确，也不“合格”，原因是算法低效、思路不清、可读性差等。实际上，在理解课程内容与能够较好地完成习题之间存在着明显的差距，而算法题完成的质量与基本的程序设计的素质培养是密切相关的。因此在本书的第三部分安排了部分习题的提示或解答。对于多数有唯一确定解的题给出答案，而对算法题则有选择地作了示范解答或提示。但算法的解答都不是唯一的，我们的解答也不一定是臻于完美的。希望我们的答案或提示能起到抛砖引玉的作用，愿读者开发出更多更好的解法。热忱欢迎把这些好的算法寄给我们，以便今后再版修订，在此预先表示感谢。然而，在此我们要特别强调的是，本题集主要是为了配合高等院校的教学而编写的，因此，为了培养学生独立思考和解决问题的能力，我们要求任何个人或单位不再正式或非正式地编印出版本题集的更详尽的解答，以免干扰原书构思，恳请谅解。

本习题集的一个特点是强调规范化在算法设计基本训练中的重要地位。附录中给出了算法书写规范和实习步骤规范。教学经验表明，严格实施这些貌似繁琐的规范，对于学生基本程序设计素养的培养和软件工作者工作作风的训练将能起到显著的促进作用。为了便于模仿，附录 4 给出了实习报告的一个范例。它不仅

为实习报告的书写格式做出了样板，而且还凝聚了很多重要的和优良的程序设计思想方法、技巧、风格等。此外，类PASCAL语言的语法概要和设计意图也在附录1中作了介绍。

这本题集在力求反映程序设计和软件工程新思想方面作了很大努力，读者不难在解答、附录和其他地方发现这一点。这些新观点主要有：模块化抽象和信息隐蔽，软件生命周期，以及初步的循环不变式思想等。但是，书中不可避免地存在着谬误和有争议之处，编者诚恳地欢迎广大读者提出批评和意见，也希望能得到各种建议。

编　者

一九八七年十一月于清华园

目　录

第一部分 习 题

提 要

这部分的习题是按照教科书内容的编排而组织的。从习题一到习题十二共十二组，每一组对应于教科书中的一章。各组的题量根据对应章节教学内容的多少和重要性程度而定，几乎对教科书的每一小节都安排了对应的习题。但对每个读者来说，不必去解全部习题，而只须根据自己的情况从中选择若干求解即可。为了标明题目的难易程度，便于读者选择，我们在每个题的题号之后注了一个难度系数，难度级别从①至⑤逐渐加深，其区别大致如下：难度系数为①和②的习题是基本题，主要是为帮助读者深化理解教学内容，澄清基本概念而设；难度系数为③的习题以基本程序设计技能训练为主要目的；习题中也收纳了不少难题，其难度系数设为④和⑤，解答这些题可以激起学习潜力较大的读者的兴趣，对广泛开拓思路很有益。

应该提醒读者注意的是，不要片面追求难题。对于解难度系数为i的习题不太费力的读者，应试试难度系数为i+1的习题，但不要把太多的时间浪费在难度系数为i+2的习题上；另一方面，解答习题应注重于“精”，而不要求“多”。为此，编者在一些自认为值得向读者推荐的“好题”题号前加注了▶标记。把握住这些“关键点”，就把握住了数据结构习题、乃至数据结构课程的总脉络。

此外，习题的难度系数是一个相对量。由于读者的水平将随学习的进展而不断提高，去比较不同组别中习题的难度系数几乎

是没有意义的。

收纳的习题种类虽然不少，但不难看出，训练的重点在于算法设计技能上，这也是本书的特点。

算法是为了让人来读的，而不是供机器读的。初学者总是容易忽略这一点。算法的真正意图主要在于提供一种在程序设计者之间交流解决问题方法的手段。因此，可读性具有头等的重要性。不可读的算法是没有用的，由它得到的程序极容易产生很多隐藏很深的错误，且难以调试正确。一般地说，宁要一个可读性好、逻辑清晰简单、但篇幅较长的算法，也不要篇幅短小但晦涩难懂的算法。算法的正确性力求在设计算法的过程中得到保证，然而一开始做不到这一点也没多大关系，可以逐步做到。

算法设计的正确方法是：首先理解问题，明确给定的条件和要求解决的问题，然后按照自顶向下，逐步求精，分而治之的策略逐一地解决子问题，最后严格按照附录2提供的算法书写规范，使用附录1提供的类PASCAL语言完成算法的最后版本。

最后要提醒读者的是，按照规范书写算法是一个值得高度重视的问题。在基础训练中就贯彻这一规范，不但能够有助于写出“好程序”，避免形成一系列难以纠正且遗害无穷的程序设计坏习惯，而且能够培养软件工作者应有的严谨的科学工作作风。

习题一（预备知识）

教科书第一章〈绪论〉的基本内容是：数据和数据结构等名词和术语的确定含义；描述算法的类PASCAL语言及从时间和空间角度分析算法的方法。

这一章的学习要点是：1.熟悉各名词、术语的含义，掌握基本概念，特别是数据的逻辑结构和存储结构之间的关系。分清哪些是逻辑结构的性质，哪些是存储结构的性质；2.熟悉类PASCAL语言的书写规范，特别要注意值参和变参的区别，输入、输出的

方式以及错误处理方式；3.了解算法五要素：①动态有穷性（能执行结束）；②确定性（对于相同的输入执行相同的路径）；③有输入；④有输出；⑤可行性（用以描述算法的都是足够基本的运算）；4.掌握计算语句频度和估算算法时间复杂度的方法。

为有助于本章的学习，在习题一中安排了15道习题，其中第1至5题帮助读者复习教科书的正文内容；第6至10题的目的是使读者通过编写算法学习使用类PASCAL语言中各种数据类型、基本语句、过程和函数及其参数表的用法。读者应特别注意熟悉记录型变量、情况语句的正确使用方法、值参和变参的区别等，对于已经熟练运用PASCAL语言的读者可跳过这些题目；第11题至14题在于帮助读者掌握计算语句频度的方法和深入理解算法时间复杂度的含义。

1.1① 简述下列术语：数据、数据元素、数据对象、数据结构、存储结构和数据类型。

1.2② 试描述数据结构的概念与程序设计语言中数据类型概念的区别。

▶1.3② 设有数据结构（D,R），其中$D=\{d_1,d_2,d_3,d_4\}$，$R=\{r\}$，$r=\{(d_1,d_2),(d_2,d_3),(d_3,d_4)\}$。试按图论中图的画法惯例画出其逻辑结构图。

1.4② 试比较在附录1中给出的算法描述语言与标准PASCAL语言的差异。

1.5② 试画出与下列程序段等价的框图。

（1）
```
product:=1; i:=1;
WHILE i≤n DO [product:=product*i;
              i:=i+1];
```

(2) i:=0;
REPEAT i:=i+1 UNTIL (a[i]=x) OR(i=n);
(3) CASE
x<y: z:=y-x;
x=y: z:=ABS(x*y)⊖;
ELSE z:=(x-y)/ABS(x)*ABS(y)
END;

1.6② 试写一算法，自大至小依次输出顺序读入的三个整数X，Y和Z的值。

1.7③ 已知k阶斐波那契序列的定义为

$f_0=0$，$f_1=0$，…，$f_{k-2}=0$，$f_{k-1}=1$；

$f_n=f_{n-1}+f_{n-2}+\cdots+f_{n-k}$，$n=k,k+1$，…

试编写求k阶斐波那契序列的第m项值的函数算法，k和m均以值参的形式在参量表中出现。

1.8③ 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

试编写算法，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。

▶1.9⑤ 试编写算法计算i!*2^i的值并存入数组a (1:arrsize)的第i个分量中 (i=1,2,…,n)。假设计算机中允许的整数最大

⊖ ABS为取绝对值函数.

值为maxint，则当n>arrsize或对某个k(1≤k≤n) 使k!*2^k>maxint时，应按出错处理。可有下列三种不同的出错处理方式：

（1）用ERROR语句终止执行并报告错误；

（2）用布尔函数实现算法以区别正确返回或错误返回；

（3）在过程的参数表中设置一整型变量以区别正确返回或某种错误返回。

试讨论这三种方法各自的优缺点，并以你认为较好的方式实现之（注意：n为值参，a为变参）。

▶1.10④ 试编写算法求一元多项式$P_n(x)=\sum_{i=0}^{n}a_ix^i$的值$P_n(x_0)$，并确定算法中每一语句的执行次数和整个算法的时间复杂度。注意：本题中的输入为$a_i(i=0,1,\cdots,n)$，x_0和n，输出为$P_n(x_0)$。通常算法的输入和输出可采用下列三种方式之一：

（1）通过read和write语句；

（2）通过参数表中的参数显式传递；

（3）通过全局变量隐式传递。

试讨论这三种方法的优缺点，并在本题算法中以你认为较好的一种方式实现输入和输出。

1.11④ 设n为正整数。试确定下列各程序段中带标号@的语句的频度：

```
(1) i:=1; k:=0;
    WHILE i≤n-1 DO【@k:=k+10*i; i:=i+1】;
(2) i:=1; k:=0;
    REPEAT  @k:=k+10*i;  i:=i+1
    UNTIL NOT(-i≤n-1);
(3) i:=1; k:=0;
    WHILE i≤n-1 DO【i:=i+1; @k:=k+10*i】;
```

▶（4）
```
FOR i:=1 TO n DO
    FOR j:=1 TO i DO
      FOR k:=1 TO j DO @x:=x+delta;
```

（5）
```
i:=1; j:=0;
WHILE i+j≤n DO
  @IF i>j THEN j:=j+1
          ELSE i:=i+1;
```

▶（6）
```
x:=n; {n是不小于1的常数}y:=0;
WHILE x≥(y+1)*(y+1) DO @y:=y+1;
```

▶（7）
```
x:=91; y:=100;
WHILE y>0 DO
  @IF x>100 THEN [x:=x-10; y:=y-1]
            ELSE x:=x+1;
```

1.12② 按增长率由小至大的顺序排列下列各函数：
2^{100}，$(3/2)^n$，$(2/3)^n$，$(4/3)^n$，n^n，$n^{3/2}$，$n^{2/3}$，$\sqrt{n}$，
$n!$，n，$\log_2 n$，$n/\log_2 n$，$\log_2^2 n$，$\log_2(\log_2 n)$，
$n\log_2 n$，$n^{\log_2 n}$。

1.13③ 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秒（一百多天），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。

1.14③ 试设定若干n值，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。

1.15③ 试用数学归纳法证明：

(1) $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6 \quad (n \geqslant 1)$

(2) $\sum_{i=0}^{n} x^i = (x^{n+1}-1)/(x-1) \quad (x \neq 1,\ n \geqslant 0)$

习题二（线性表）

教科书第二章〈线性表〉的基本内容是：线性表的逻辑结构定义和各种存储结构的描述方法；在线性表的两类存储结构（顺序的和链式的）上如何实现基本运算，及用线性表表示稀疏多项式时如何实现稀疏多项式的相加。

这一章的学习要点是：1.了解线性表的逻辑结构特性是数据元素之间存在着线性关系，在计算机中表示这种关系的不同方法得到两类不同的存储结构；2.熟练掌握这两类存储结构的描述方法，如一维数组中一个区域[i..j]的上、下界和长度之间的变换公式($\mathscr{L}=j-i+1$，$i=j-\mathscr{L}+1$，$j=i+\mathscr{L}-1$)，链表中指针p和结点p↑的对应关系（结点p↑.next↑是结点p↑的后继等），链表中的头结点、头指针和首元结点的区别及循环链表、双重链表的特点等；3.熟练掌握线性表在顺序存储结构上实现基本运算：查找、插入和删除的算法；4.熟练掌握在各种链表结构中实现线性表运算的基本方法，并学会何时选用何种链表的方法；5.掌握从时间和空间复杂度的角度综合比较线性表两种存储结构的不同特点及其适用场合。

和这一章的要求相匹配，在习题二中安排了难度渐增的五类习题，第一类只涉及线性表在顺序结构上各种基本运算的实现；第二类涉及线性链表的各种运算；第三类涉及两个或多个线性表的各种运算；第四类对不同的存储结构作对照比较，并注重其时间复杂度的分析；第五类涉及循环链表、双重链表和稀疏多项式

在线性表的两种存储结构上运算的实现。

2.1① 描述以下三个概念的区别：头指针，头结点，首元结点（第一个元素结点）。

2.2① 指出以下算法中的错误和低效(即费时）之处，并将它改写为一个既正确又高效的算法。

```
PROC delk (a: ARRAY[1..arrsize] OF integer;
           last, i,k:integer);
{a[1..last]含线性表元素。本过程从a[1..last]中删除第i个
元素起的k个元素}
IF(i<1)OR(i>last)OR(k<0)OR(last>arrsize)
  THEN ERROR('Arguement invalid')
  ELSE {Arg.valid}
       FOR count:=1 TO k DO{删一个元素}
         [FOR j:=last DOWNTO i+1 DO
              a[j-1]:=a[j];
          last:=last-1
         ]
ENDP; {delk}
```

▶2.3② 设线性表存于a(1:arrsize)㊀的前 elenum 个分量中，且递增有序㊁。试写一算法，将 x 插入到线性表的适当位置上，以保持线性表的有序性。

㊀ 在本集中将一致以arr(1:arrsize)（arr为数组名）表示一个数组空间，而以arr[i..j]或arr[k]表示存于该空间中的元素。

㊁ 在本集中，凡递增有序即非递减有序，对具有此类性质的结构中的元素，不妨设为整型。

▶2.4③ 设$A=(a_1,\cdots,a_m)$和$B=(b_1,\cdots,b_n)$均为有序线性表，A′和B′分别为A和B中除去最大共同前缀后的子表（例如，A=(x,y,y,z,x,z)，B=(x,y,y,z,y,x,x,z)，则两者的最大共同前缀为x,y,y,z,在两表中除去最大共同前缀后的子表分别为A′=(x,z)，B′=(y,x,x,z))。若A′=B′=空表，则A=B；若A′=空表,B′≠空表，或两者均不空且A′首元小于B′首元，则A<B；否则A>B⊖。试写一个比较A、B大小的算法。

2.5② 试写在带头结点的动态单链表结构上实现线性表操作LOCATE(L,X)和LENGTH(L)的算法。

2.6② 已知指针ha和hb分别指向两个单链表的头结点，且头结点的数据域中存放链表的长度（链表中的元素均为整数）。试写一算法PROC connect(ha, hb:linklist;VAR hc:linklist)：将这两个链表连接在一起（即令其中一个表的首元结点连在另一表的最后一个结点之后），hc指向连接后的链表的头结点，并要求算法以尽可能短的时间完成连接运算。请分析你的算法的时间复杂度。

2.7③ 已知指针la和lb分别指向两个无头结点单链表中的首元结点。下列算法是从表la中删除自第i个元素起共len个元素后，并将他们插入到表lb中第j个元素之前。试问此算法是否正确？若有错，则请改正之。

```
PROC insertsub(VAR la,lb:linklist; i,j,len: integer);
  IF(i<0)OR(j<0)OR(len<0)
    THEN ERROR('Arguement invalid');
```

⊖ 这里定义的两线性表间比较大小的方法即定义了在线性表的集合上的一个全序关系，即词典次序.

```
    p:=la; k:=1;
    WHILE k<i DO 【p:=p↑.next; k:=k+1】;
    q:=p;
    WHILE k≤len DO 【q:=q↑.next; k:=k+1】;
    s:=lb; k:=1;
    WHILE k<j DO 【s:=s↑.next; k:=k+1】;
    s↑.next:=p; q↑.next:=s↑.next
ENDP; {insertsub}
```

2.8② 试写在无头结点的动态单链表上实现线性表操作INSERT(L,i,b)和DELETE(L,i)的算法，并和在带头结点的动态单链表上实现相同操作的算法进行比较。

▶2.9③ 已知线性表中的元素以值递增有序排列，并以单链表⊖作存储结构。试写一高效的算法，删除表中所有值大于mink且小于maxk的元素（若表中存在这样的元素），并分析你的算法的时间复杂度（注意：mink和maxk是给定的两个参变量，它们的值为任意的整数）。

2.10② 同上题条件，试写一高效的算法，删除表中所有值相同的多余元素（使得运算后的线性表中所有元素的值均不相同），并分析你的算法的时间复杂度。

▶2.11③ 试分别以不同存储结构实现线性表的就地逆置算法，即在原表的存储空间内将线性表 $(a_1,a_2\cdots,a_n)$ 逆置为$(a_n, a_{n-1},\cdots,a_1)$。

（1）以一维数组作存储结构，设线性表存于a（1:arrsize）

⊖ 今后若不特别指明，则凡以链表作存储结构时，均带头结点。