数据结构实验与学习指导书

徐孝凯 等编

中央广播电视大学出版社

数据结构实验与学习指导书

徐孝凯　等编

中央广播电视大学出版社

(京)新登字 163 号

图书在版编目(CIP)数据

数据结构实验与学习指导书/徐孝凯等编. —北京:中央广播电视大学出版社,1996.10
ISBN 7-304-01265-X

Ⅰ.数… Ⅱ.徐… Ⅲ.数据结构-实验-电视大学-教学参考资料 Ⅳ.TP311.12-33

中国版本图书馆 CIP 数据核字(96)第 21511 号

数据结构实验与学习指导书
徐孝凯 等编

中央广播电视大学出版社出版
社址:北京市复兴门内大街 160 号 邮编:100031
北京印刷二厂印刷 新华书店北京发行所发行
开本 787×1092 1/16 印张 17.75 千字 443
1996 年 10 月第 1 版 1996 年 10 月第 1 次印刷
印数:1—20000 册
定价 19.55 元
ISBN 7-304-01265-X/TP·66

前　言

《数据结构》是计算机各专业的一门核心课题，主要研究数据的逻辑结构、存储结构以及相应运算的算法。数据的逻辑结构概括为线性结构、树型结构和图型结构三大类。线性结构较简单，数据元素之间是1对1的关系；树型结构较复杂，数据元素之间是1对N的关系；图型结构则更加复杂，数据元素之间是M对N的关系。数据的存储结构概括为顺序结构、链接结构、散列结构和索引结构四大类，每一类都具有各自的优点和缺点，都分别适应于不同的应用场合。对数据的基本运算概括为查找、插入、删除、排序和遍历等，当数据的逻辑结构、存储结构及程序设计方法不同时，相应运算的算法也就不同。

本书是由中央广播电视大学出版社出版的、由江涛主编的《数据结构》一书配套使用的辅助教材，它包括实验、学习指导和附录三个部分的内容。实验部分共给出了七个实验，以Turbo Pascal语言为运行环境。学习指导部分组织成16讲，主要对树、图、查找及排序等方面的概念和算法进行了详细的分析和讨论，第16讲对整个课程提出了具体教学要求。附录部分包含三个方面的内容：一是给出了主教材中绝大部分习题的参考答案，二是给出了该门课程经修改后的教学大纲，三是给出了四套已经使用过的数据结构试题及答案，供同学们期末复习时参考。

本书由中央广播电视大学徐孝凯、武汉市广播电视大学左俊明、江西省广播电视大学钟声和云南省广播电视大学解季萍共同编写，其中左俊明编写前四个实验，钟声编写后三个实验，解季萍编写学习指导部分的前五讲，徐孝凯编写其余部分。由于编者水平有限，加之时间仓促，错误和不当之处在所难免，希望广大读者批评指正。

本书除作为中央广播电视大学统设的“数据结构”课程的辅助教材外，亦可供讲授和学习“数据结构”课程的广大师生参考。

编　者

1996年5月

目　录

实验部分

学习指导部分

附录部分

实 验 部 分

实验一 线 性 表

一、实验目的

1. 掌握用 Turbo Pascal 调试程序的基本方法。

2. 掌握线性表的基本运算，如插入、删除、进栈、出栈、队列及串在顺序存储结构上的实现。

二、实验预备知识

1. Turbo Pascal 简介

Turbo Pascal 具有编译速度快，运行效率高，用户界面好，即菜单驱动等特点。除了实现标准 Pascal 的所有功能之外，还有很多扩充功能。Turbo Pascal 有一个称为集成环境的主工作系统，它集编辑、编译、连接、调试运行于一体，是一个功能强大而又方便的工作系统。用户可在不退出系统的情况下完成所有的操作。Turbo Pascal 由美国 Borland 公司推出，问世以来其版本不断更新，它可以在多种操作系统下运行，广泛应用于 IBM PC 及其兼容机上。

2. Turbo Pascal 的组成

Turbo Pascal 3.01 版是初学者容易掌握的一个版本，也是学习更高级版本的基础，它由以下几个文件组成。

(1) TURBO · COM 编译编辑程序

当你在终端上键入命令 turbo ↵ 时，这个文件被调入内存并开始运行。它是 Turbo Pascal 的主程序，包括编辑器和编译器等。

(2) TURBO · MSG 错误信息文本文件

Turbo Pascal 启动后，系统会提问：

Include error messages (Y/N)?

若回答 Y，这个文件被调入内存，且当编译出错时，可看到关于错误解释的文字信息。若回答 N，编译出错时 仅给出错误编号，此时可查阅手册中的错误信息表，得到适当的解释。通常回答 N，这样可节省约 1.5kB 的内存空间。

(3) TINST · COM 安装程序

启动 Turbo Pascal 前，在键盘上打入 TINST 命令，便可进入安装过程，屏幕上出现一个窗口式菜单，用户可根据需要，选择适当的参数或设备，根据菜单中的提示，自动安装。

(4) TINST·MSG 安装信息文本文件

该文件可提供安装过程中的文本信息。

(5) GRAPH·P 绘图文件

当你需要绘图时,可把此文件装在运行盘中,它包含使用扩展图形以及 GRAPH·BIN 中包含的龟绘图子程序所必须的外部说明。

(6) GRAPH·BIN 绘图子程序

该子程序包含扩展图形以及龟绘图的机器代码子程序,供用户绘图时使用。

3. Turbo Pascal 的安装和启动

Turbo Pascal 3.01 版本的全部文件装在一张软盘中,因而既可以在软盘中使用,也可把它安装到硬盘中使用。

(1) 用软盘使用时只需将装有该系统的磁盘插入驱动器 A 或者 B,关好门,从键盘上键入 turbo ↵即可。

(2) 在 C 盘中使用时,先建立 TP 子目录,然后将软盘中的全部文件拷贝到该目录下,然后键入 turbo ↵即可。

4. Turbo Pascal 的使用

当键入 turbo ↵后,片刻屏幕显示如下信息:

```
TURBO Pascal System Version 3.01A
                    PC-DOS
Copyright(C) 1984,85 BORLAND Inc.

Default display mode
Include error messages  (Y/N)? _
```

线框中的内容为 Turbo Pascal 的版本号及版权信息,"Default display mode"表示当前显示方式为缺省方式。"Include error messages (Y/N)?"询问是否要装入错误信息?回答"Y",表示错误信息文件将被读入内存(要求它在运行盘上)并显示信息:

Loading TURBO·MSG

在今后的编译、连接过程中,如果出现错误,系统将自动地给出错误类型提示,以便改正程序中的错误,提高调试程序的质量。如果用户想节约内存,则回答"N"。这时被编译的程序出错,只给出错误代号,不显示提示,要想得到提示,可根据代号查阅其对应的错误信息表。随后屏幕上出现系统主菜单:

在上述菜单中,大写字母以高亮度显示,使用任何命令,只要键入相应的大写字母即可。

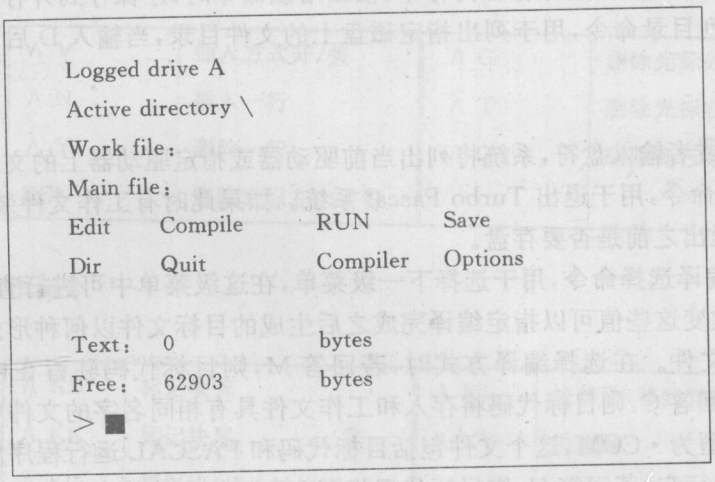
(1) Logged drive 是驱动器选择,标记当前使用的驱动器提示符,如需改变,则选择 L 命令,键入 L 后,将出现以下提示:

New drive: _

此时输入一个新驱动器的名称及回车。

(2) Active directory 是路径选择,标记当前使用的路径,如果需改变当前路径用 A 命令,

输入 A 命令后屏幕将出现以下提示符：



New directory：

此时输入一个新的路径名即可

(3) Work file 是工作文件的选择，即用于编辑、编译、执行及存储的文件。当键入 W 命令时，屏幕上出现以下提示：

Work file name:-

此时将你要编辑、修改或执行的文件名输入即可。如果输入的文件没有扩展名，系统会自动加上扩展名·PAS，若输入的文件名是已有的，计算机将把该文件从磁盘上读入内存，并显示如下提示：

Loading C:\TP\EXAM·PAS

其中 EXAM·PAS 是用户输入的文件名，如果磁盘上没有你输入的名字的文件，将显示如下提示：

Loading C:\TP\EXAM·PAS

New File:

其中 C:表示当前磁盘，\TP\表示当前路径，EXAM·PAS 表示当前工作文件，New file 表示这是一个新文件。

(4) Main file 是主文件的选择，Turbo Pascal 允许将一个大程序分成几个小程序单独地进行编写，在编写中通过一个主文件对其它文件进行调用。

(5)Edit 是编辑器，Turbo Pascal 的集成环境下有一个内部编辑器，其编辑命令是 WS 的子集，这对于熟悉 WS 的用户是非常方便的。

(6)Compile 是编译程序，如果没有说明主文件，将对工作文件进行编译，否则编译主文件。如果工作文件编辑完毕，在主文件装入编译之前，系统将要询问是否要存盘，按任意键将中止编译过程。

(7)Run 是运行命令，用于执行内存中的程序或者·COM 文件。如果内存中的程序是一个已编译的文件，则可直接运行。若内存中的程序事先没编译，当输入 R 命令后，先进行编译，然后再运行。

(8)Save 存储命令，用于把当前编辑过的工作文件存储到磁盘上，Turbo Pascal 中编辑器在退出编辑并不自动地存盘，而是保存在内存中，仅当用该命令时，才保存到外存中。

(9)Directory 是列目录命令，用于列出指定磁盘上的文件目录，当输入 D 后，屏幕上将显示如下信息：

DIR MASK：

此时可打回车键或者输入盘符，系统将列出当前驱动器或指定驱动器上的文件信息。

(10)Quit 是退出命令，用于退出 Turbo Pascal 系统。如果此时有工作文件装入且已经编辑过，系统将询问在退出之前是否要存盘。

(11)Options 是编译选择命令，用于选择下一级菜单，在这级菜单中可进行查看和规定编辑器的缺省值，通过改变这些值可以指定编译完成之后生成的目标文件以何种形式存放，是内存文件还是·COM 文件。在选择编译方式时，若回答 M，则目标代码驻留在内存中，等待 RUN 命令启动它；若回答 C，则目标代码将存入和工作文件具有相同名字的文件中(或存入主文件中)，并且文件类型为·COM，这个文件包括目标代码和 PASCAL 运行程序库，在终端上键入它的名字，即可运行它；若回答 H，则目标代码将存入和工作文件具有相同名字的文件中，或者存入主文件中，其文件类型为·CHN.

5. 编辑器的使用

Turbo Pascal 编辑器是一个全屏幕的编辑器，当你指定了工作文件之后，键入 E 命令便进入编辑状态，如果工作文件在登录驱动器上，便自动装入工作文件并显示正文的第 1 页。如果是新文件，则除了最上面一状态行外，屏幕都是空白的，可供你输入源程序文件。

(1) 状态行包括以下信息：

Line n Col n Insert Indent A:EXA·PAS

Line n 表示光标所在行的行号，行号 n 从文件首开始计算。

Col n 表示光标所在列的列号，列号 n 从光标所在行最左边开始计算。

Insert 表示当前是处在插入状态还是修改状态(Insert/Overwrite)两种状态可轮番转换。

Indent 表示自动缩排有效，为一开/关命令。

A:EXA·PAS 表示正在编辑的文件的驱动器，文件名和扩展名。

(2)编辑命令一览表

① 光标移动命令

∧ S 或 ←	左移一个字符	∧ R	上移一页
∧ D 或 →	右移一个字符	∧ C	下移一页
∧ A	左移一个字符	∧ QS	移到行的左边
∧ F	右移一个字符	∧ QD	移到行的右边
∧ E 或 ↑	上移一行	∧ QE	移到屏幕顶部
∧ X 或 ↓	下移一行	∧ QX	移到屏幕底部
∧ W	上滚	∧ QR	移到文件开始
∧ Z	下滚	∧ QC	移到文件末尾

② 插入和删除命令

∧ V	插入方式开/关	∧ G	删除光标处字符
∧ N	插入一行	∧ T	删除光标右边字符
∧ Y	删除一行	∧ QY	删除到行尾
DEL	删除光标左边字符		

③ 块操作命令

∧ KB	标记块首	∧ KV	块移动(移到光标处)
∧ KK	标记块尾	∧ KY	块删除
∧ KC	复制块(将作过块标记的块复制到光标处)	∧ KW	写一个块到磁盘上
		∧ KR	从磁盘上读一个块

④ 其它操作命令

∧ KD	结束编辑	∧ QL	恢复行
∧ L(或 TAB)	制表	∧ QF	查找字符串
∧ QI	自动缩排开/关	∧ QA	查找并替换字符串

6. 程序的调试

Turbo Pascal 程序的调试分为下列三个步骤：

第一步　利用程序编辑器 Edit 输入源程序，并进行编辑修改，直到你认为没有错误为止。然后退出编辑状态回到主菜单。

第二步　在 Turbo Pascal 主菜单下，选择 C 命令，对当前内存中的程序进行编译，此时如果发现程序行中有错误，Compiling 便会指出错误所在的行，并给出错误代号或说明信息，此时再按 ESC 键返回到编辑状态，光标自动停留在错误处，利用 edit 编辑键改正错误，并存盘返回到主菜单，再选择 C 命令重新编译，重复进行第一、二两步直到屏幕上出现下列信息为止。

Compiling

n line

Code:……

Data:……

Stack/Heap……

这表明编译已经完成，其中 n line 表示程序共有 n 行，Code 表示程序代码在内存中的地址，Data 为数据地址，Stack 为堆栈地址等。如果要运行该程序，只需输入 R 命令。

第三步　在主菜单中选择 R 命令运行内存中的程序，此时屏幕上立刻会显示运行后的结

果。

如果结果与预计的值相差甚远，则要重新检查程序中是否有逻辑错误。

7. TURBO PASCAL crt 单元一些过程和函数的功能介绍。

CLRSCR：清屏幕；

GOTOXY(i,j)：光标移动到屏幕的 j 行 i 列；

UPCASE(ch)：将小写字母转换成大写；

READLN(ch)：对 5.0 版相当于 Read(ch);Readln；

TEXTBACKGROUND(N)：设置屏幕底色，N=1 为蓝色；

TEXTCOLOR(N)：设置字符颜色，N=4 为红色，N=15 为白色；

WRITE(^G)：响铃一次；

filesize(f)：返回文件 f 的记录数；

Seek(f,N)：指针移到文件 f 的第 N 条记录。

其它过程、函数本实验不使用，在此不作介绍，有兴趣的同学请参阅《用户参考手册》。实验五、六、七使用了以上过程，并在 5.0 版基础上调试通过。

三、实验内容

1. 线性表在顺序存储结构上的插入运算。

(1) 问题描述

当我们要在线性表第 i 个位置上插入或者删除一个元素时，必须先将线性表中原有的第 i 个元素之后的所有元素依次后移一个位置，以便腾出一个空位置 i，再把新元素存入到该位置上，如果删除线性表中第 i 个元素，也必须将第 i 个元素之后的所有元素前移一个位置。

(2) 基本要求

设计一个插入算法的程序，将一个指定的数据插入到一个线性表的指定位置上。

设一个线性表是有序的，其中有 8 个数据元素，如图所示。

序号	1	2	3	4	5	6	7	8	
线性表	12	13	21	24	28	30	42	77	

插入前 n=8

现要插入一个值为 25 的元素，并要求插入该元素之后，线性表仍是有序的。

(3) 实现提示

上图线性表中的 8 个有序(升序)元素，每个数据元素占用一个存储单元，插入的 25 必须在 24 和 28 两元素之间，故从 28 开始，以后的每个元素必须后移一个位置，如下图所示。

序号	1	2	3	4	5	6	7	8	9
线性表	12	13	21	24		28	30	42	77

① 若第 8 个元素之后没有空闲的位置，这时插入一个新元素，则会发生上溢出，此时程序要给出上溢出信息：Over flow

解决上溢出的办法是寻找一个较大的空间，然后将线性表中的元素依次复制到新空间内，再进行插入。

② 插入后线性表如下图所示。

序号	1	2	3	4	5	6	7	8	9	
线性表	12	13	21	24	25	28	30	42	77	插入之后 n=9

(4) 实验程序

```
PROGRAM SHIYAN1·1 (input, output);
  TYPE
    atype=array[1··10)] of real;
  VAR
    list: atype;
    n,i,j: integer;
    x: real;
PROCEDURE insert (var list: atype; var n, i: integer; var x);
  BEGIN
    for j:=n downto i do
    list[j+1]:=list [j];
    list[i]:=x;
    n:=n+1;
  END
BEGIN
    write ('please inter the integer number:');
    readln(n);
    for j:=1 to n do
      readln ( list [j]);
    write ('please inter i and x');
    readln (i,x);
    while (i<1) or (i>=n) do
      BEGIN
        write ('please inter i and x again:');
        readln (i,x);
      END
    insert (list,n,i,x);
    for j:=1 to n do
    write ( list [j]);
  END
```

2. 线性表在顺序存储结构上的删除运算。实验程序如下：

```
PROGRAM SHIYAN1・2 (input, output);
  TYPE
    atype=array[1・・10)] of real;
  VAR
    list: atype;
    n,i,j: integer;
PROCRDURE delete (var list: atype; var n, i: integer);
  BEGIN
    for j:=i to (n-1) do
    list[j]:=list [j+1];
    n:=(n-1);
  END;
BEGIN
  write ('please inter the integer number:');
  readln(n);
  for j:=1 to n do
  readln ( list [j]);
  write ('please inter i;');
  readln (i);
  while (i<1) or (i>n) do
      BEGIN
        write ('please inter i again:');
        readln (i);
      END;
    delete (list,n,i);
    for j:=1 to n do
    write ( list [j]);
  END
```

四、实验要求

1. 编写对有 N 个元素的线性表进行插入和删除的算法，并判断插入和删除的位置是否超出范围。

2. 根据给出的实验源程序进行调试运行，并写出输入、输出的结果和溢出判断结果。

3. 写出上机调试运行后的体会。

实验二　链　接　表

一、实验目的

(1) 进一步掌握 Turbo Pascal 的操作方法，熟练地运用编辑器 Edit 或者 WS 来建立和修改源程序文件。

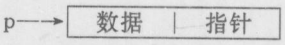
(2) 尽量减少依赖于具体机器计算机环境的用法，充分利用 Pascal 语言的结构化特点，利用其递归能力，提高程序的可移植性。

(3) 掌握链接表的基本运算，如插入、删除、进栈、出栈及队列在链式存储结构上的实现。

二、实验预备知识

从实验一我们体会到线性表的顺序存储结构在逻辑关系上和物理位置上两个元素均是相邻的，简单、直观。但其在插入删除操作时，需移动大量元素，如果线性表长度变化大，则预存空间也大，线性表的容量也难以扩充。

链式存储结构是用任意的存储单元存储表的数据元素，它包括两个域，其中存储数据元素信息的域称为数据域，存储直接后继存储位置的域称为指针域，其表示法如图所示：



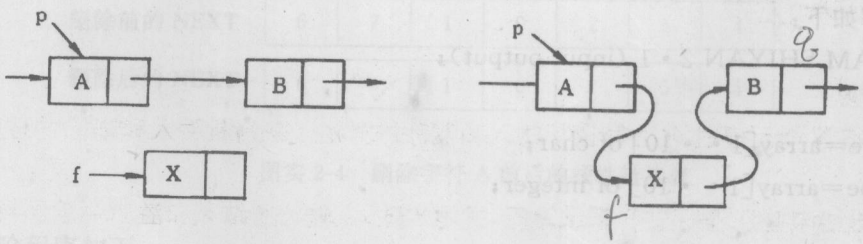
数据域 data　　指针域 next

p 为指针变量，其值是它所指结点的首地址。

p↑·data 表示 p 所指结点数据域中的数据。

p↑·next 表示 p 所指结点指针域中的指针，通常为下一个结点的首地址，也可能为空。

插入和删除的基本操作如图实 2-1 和 2-2 所示。

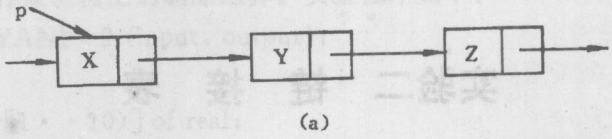


(a) 插入新元素 X 之前　　(b)插入新元素 X 之后

图实 2-1　插入一个结点到链表中

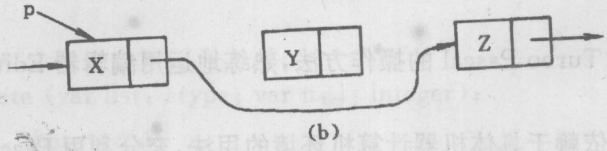
设指针 p 指着 A 结点，指针 f 指着将要插入的新结点 X，X 插在线性表两个数据元素 A 和 B 之间，这时只要修改两个指针即可，其语句描述为

f↑·next:=p↑·next;　p↑·next:=f;



(a)

(a) 结点 Y 删除之前



(b)

(b) 结点 Y 删除之后

图实 2-2　从链表里删除一个结点

图中 p 为指向结点 X 的指针，若要删除链表中一个元素 Y，则仅修改一个指针，可用语句描述为

p↑ · next:=p↑ · next↑ · next

三、实验内容

1. 已知有一线性链接表，如图所示，现要将一下标为 8 的新元素 T 插入到下标为 2 的结点 U 之后，使原结点顺序 C O M P U E R 变为 C O M P U T E R.

	1	2	3↓f	4	5	6	7	8
DATA	O	U	C	R	P	M	E	T
插入前的 NEXT	6	7	1	0	2	5	4	
插入后的 NEXT	6	8	1	0	2	5	4	7

图实 2-3　插入字符 T 前、后的线性链接表

实验程序如下：

```
PROGRAM SHIYAN 2 · 1 (input output);
  TYPE
    atype=array[1 · · 10] of char;
    btype=array[1 · · 10] of integer;
  VAR
    DATA: atype;
    NEXT: btype;
    i,p,n: integer;
PROCEDURE traversal (F:integer);
  BEGIN
    P:=F
```

```
    N:=0
    while P<>0 do
BEGIN
    write (data [p]);
      p:=next [p];
      n:=n+1
    END
    writeln;
    writeln(n);
  END
BEGIN
  for i:=1 to 7 do
    BEGIN
      READLN( data [i], next [i]);
    END
  traversal (3);
      data [8]:='T';
      next [8]:=next [2];
      next [2]:=8;
      traversal (3);
END
```

2. 已知有一线性链接表如图所示，现要将一下标 i=2 的元素 A 删除，使各结点顺序由原来的 D E S I A G N 变为 D E S I G N.

	1	2	3↓f	4	5	6	7
DATA	E	A	D	N	I	S	G
删除前的 NEXT	6	7	1	0	2	5	4
删除后的 NEXT	6	⋯	1	0	7	5	4

图实 2-4　删除字符 A 前后的线性链接表

实验程序如下：

```
PROGRAM SHIYAN 2·2 (input output);
  TYPE
    atype=array[1··10] of char;
    btype=array[1··10] of integer;
  VAR
```

```
    data: atype;
    next: btype;
    i,p,n: integer;
PROCEDURE traversal (f:integer);
  BEGIN
    P:=f;
    n:=0;
    while P<>0 do;
      BEGIN
        writrln (DATA [P]);
        P:=next [p];
        n:=n+1;
      END
    writeln;
    writeln(n);
  END
BEGIN
  for i:=1 to 7 do
  readln (data [i],next [i];
  traversal (3);
  write ('please inter the i:');
  readln (i); {删除 i 结点的后继结点,如 i=5,则删除下标为 2 的结点}
    next [i]:=next [next [i]];
    traversal (3);
END.
```

四、实验要求

1. 编写算法,对具有 n 个元素的线性链接表中指定的元素之前进行插入运算;删除线性链接表中指定元素的结点。

2. 根据给出的实验程序进行调试运行,写出输入和输出结果。

3. 写出调试运行的分析和体会。