

数据结构

习题集与解题指导

薛晓燕 王晓冬 编著

444

科学技术文献出版社

数据结构习题集与解题指导

薛晓燕 王晓冬 编著

科学技术文献出版社

(京)新登字 130 号

内 容 简 介

本书是为配合计算机有关专业开设“数据结构”课程而编写的教学参考书。习题部分以大量不同难度的问题帮助读者从整体上掌握“数据结构”的主要内容,提高综合解题能力。每一章选出典型习题作出详细解答,其余部分给出参考答案或解题思路。试题部分收集了近年来函授数据结构试题以及部分高校硕士研究生数据结构入学试题,并给出参考答案。书中所列算法程序均在 PC 机上用 Turbo Pascal 5.0 以上版本通过,给出程序清单及运行结果,并可提供源程序软盘。

本书不仅可作为大专院校计算机专业的辅导教材,也可作为从事计算机方面工作的广大科技人员及函授、电大、夜大读者的参考书,对于准备参加计算机专业硕士研究生入学考试的学生来说是一本很好的复习材料。

图书在版编目(CIP)数据

数据结构习题集与解题指导/薛晓燕等编著.

北京:科学技术文献出版社,1995.7

ISBN 7-5023-2475-5

I. 数… I. ①薛… ②王… III. 数据结构—习题—题解
IV. TP311.12

中国版本图书馆 CIP 数据核字(95)第 01703 号

科学技术文献出版社出版

(北京复兴路 15 号 邮政编码 100038)

北京建外印刷厂印刷 新华书店北京发行所发行

1995 年 7 月第 1 版 1995 年 7 月第 1 次印刷

787×1092 毫米 16 开本 20.125 印张 514 千字

科技新书目:356-097 印数:1-2500 册

定价:28.00 元

前言

“数据结构”是计算机系各专业的重要基础课程。该课程通过介绍最常用的数据结构,阐明数据结构的逻辑关系,在计算机内的存储表示,以及相应结构下的算法,培养学生根据实际问题选择合理数据结构与有效算法的技能。为了学好该课程,必须通过大量的习题和上机练习,来巩固和体会所学知识,提高软件设计水平和编程水平。本书正是配合“数据结构”编写的,既可作为例题讲授,又可作为作业布置,还可作为自学教材使用。

本书所列习题都是编者精心选择的。编者多年来讲授“数据结构”课程,批改作业,辅导实验,指导优秀学生准备硕士研究生考试,积累了宝贵的资料和经验。本书参考了近年来国内外出版的数据结构方面的教科书及文献,习题丰富,内容新颖。

本书共分为四大部分。第一部分是习题集,由八章组成,包含 267 个习题。每章首先给出

分数	解 释
00	心算题与基本概念题
10	简单,5~10 分钟左右
20	普通题,15~30 分钟左右
30	中等难度题,需 30 分钟~2 小时完成
40	困难或冗长的习题,适于课程设计或上机实验
50	研究题,尚未满意解决

相应理论的主要内容和学习要点,然后是习题。为了方便读者根据自己的水平选择题目,参考世界著名专家 Knuth 在其著作中采用的方法,将每个习题按难易程度标出分数:

位于两个标准分之间的习题,其难易程度也在两个标准分之间。例如分数 15,表示难度在 10~20 之间。必须说明,分数只能成为表示困难程度的一种估计,不应看作是绝对的标准。

有些习题以一个“★”开头,表示这个问题值得特别推荐。

第二部分是试题集,收集了近年来不同水平的数据结构试题。读者可根据自己的具体情况选择使用。试题可用来检验自己对本课程的总体掌握情况。

第三部分为习题答案或提示。对于有固定答案的习题列出其解。算法题分为详解、略解和提示三种。详解一般分四步完成:(1)解题思路;(2)算法描述;(3)源程序清单、输入数据、输出结果清单;(4)性能分析。略解一般分二步完成:(1)解题思路;(2)算法描述或源程序清单。

第四部分为试题参考答案。

众所周知,数据结构重在培养学生算法设计与分析的能力。一个算法题目的解答不是唯一的。这里提供的解答只能起启示和参考作用。读者在解题时,应独立思考,尽力解答。在遇到困难时,尽量只参考解题思路,以帮助解题。不要急于翻阅算法及源程序清单。或许有更好的解法,希望能及时向我们指出,以便今后再版时修正。

考虑到目前国内使用的计算机绝大多数都是 PC 及其兼容机,因此书中算法程序全部用

Pascal 语言编写,并在 PC 机上用 Turbo Pascal 5.0 以上版本调试通过。所有源程序文件都存在一张 1.2M 的高密盘上,可随书提供。

本书由山东大学计算机科学系张景淮教授审阅,在此表示衷心感谢。

由于作者知识水平有限,书中一定存在着不少缺点和错误,希望读者批评指正。

编者

1994.12

数
据
结
构
甲

B
A
B

目 录

第一部分 习题集	(1)
第一章 绪论.....	(1)
第二章 线性结构.....	(4)
第三章 数组和广义表	(12)
第四章 树结构	(14)
第五章 图结构	(20)
第六章 查找	(27)
第七章 内排序和外排序	(33)
第八章 文件系统	(38)
第二部分 试题集	(40)
试题一 数据结构函授试题(1)	(40)
试题二 数据结构函授试题(2)	(43)
试题三 山东大学 1992 年硕士研究生入学试题	(45)
试题四 山东大学 1993 年硕士研究生入学试题	(48)
试题五 北京邮电学院 1992 年硕士研究生入学试题	(50)
试题六 北京邮电学院 1993 年硕士研究生入学试题	(53)
试题七 北京邮电学院 1994 年硕士研究生入学试题	(55)
试题八 北京航空航天大学 1992 年硕士研究生入学试题	(57)
试题九 中山大学 1994 年硕士研究生入学试题	(60)
第三部分 习题答案与提示	(63)
习题一 绪论	(63)
习题二 线性结构	(70)
习题三 数组和广义表.....	(127)
习题四 树结构.....	(144)
习题五 图结构.....	(175)
习题六 查找.....	(222)
习题七 内排序和外排序.....	(249)
习题八 文件系统.....	(282)
第四部分 试题参考解答	(286)
试题一 函授试题(1)参考答案	(286)
试题二 函授试题(2)参考答案	(290)
试题三 山东大学 1992 年硕士研究生入学试题参考答案	(293)
试题四 山东大学 1993 年硕士研究生入学试题参考答案	(295)
试题五 北京邮电学院 1992 年硕士研究生入学试题参考答案	(297)
试题六 北京邮电学院 1993 年硕士研究生入学试题参考答案	(299)

试题七	北京邮电学院 1994 年硕士研究生入学试题参考答案	(301)
试题八	北京航空航天大学 1992 年硕士研究生入学试题参考答案	(304)
试题九	中山大学 1994 年硕士研究生入学试题参考答案	(307)
附录： 单元使用说明		(310)
单元一	循环单链表	(310)
单元二	循环单链表存储的多项式	(311)
单元三	二叉链表存储的二叉树	(311)
单元四	链队列(元素为二叉链表指针)	(312)
单元五	二叉线索链表存储的二叉树	(312)
单元六	树的存储结构转换	(313)
单元七	链栈(元素为二叉链表指针)	(313)
单元八	带入度的邻接表存储图结构	(314)
单元九	加权邻接矩阵存储图结构	(314)
单元十	排序用线性表	(315)

第一部分 习题集

第一章 绪 论

本章重点:

1. 有关数据结构的术语与基本概念。
2. 掌握抽象数据类型 ADT 的说明、设计和实现。
3. 熟悉类 Pascal 语言的书写规范。
4. 掌握估算时间复杂度的方法,了解空间复杂度的度量方法。

习题一 绪论

1.1 [05] 解释下列术语:

数据,数据元素,数据对象,数据类型,抽象数据类型,原子数据类型,结构数据类型,逻辑结构,存储结构,数据结构,顺序存取,随机存取,算法。

1.2 [06] 试简述算法与程序的区别。

★1.3 [10] 设有数据逻辑结构为 $\text{Data-Structure} = (D, R)$, 其中 $D = \{d_1, d_2, d_3, d_4, d_5\}$, $R = \{r\}$, $r = \{\langle d_1, d_2 \rangle, \langle d_1, d_3 \rangle, \langle d_1, d_4 \rangle, \langle d_3, d_5 \rangle, \langle d_4, d_5 \rangle, \langle d_4, d_6 \rangle\}$, 试画出其逻辑结构图。

1.4 [00] 数据结构中常用的运算有哪些? 试举出五个。

★1.5 [25] 使用抽象数据类型 ADT 有什么优点? 怎样利用 Turbo Pascal 中单元概念实现 ADT? 试举例说明。

1.6 [10] 编写一个算法,对任意给定的两个正整数 a 和 b ,求它们的最大公因数(即能够同时整除 a 和 b 的最大正整数)。

1.7 [10] 举例说明一个问题用某种算法或结构实现时十分困难,但如采用另一种算法或结构却很容易。

1.8 [10] 编写一个算法,将二个字符串按 ASCII 码顺序排列并打印。

1.9 [10] 试编写算法,对连续输入的 n 个整数,找出其中最大值和最小值(规定输入数在整数允许范围内)。

★1.10 [25] 建立某个班级的学生情况登记表。它包括班号、总人数、每一位同学的学号、姓名、性别、年龄、五门功课(英语、数学、电子学、汇编语言、数据结构)的成绩、总分和平均分。登记表最后给出该班的平均年龄,各门功课的平均成绩,总分及平均分。假定人数不超过 30 人。

1.11 [20] 编写一个算法,求一元多项式 $A(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ 在 x_0 处的值 $A(x_0)$,并确定该算法的时间复杂度。

★1.12 [10] 设 n 为正整数,分析下列各程序段的时间复杂度:

(1) procedure matrix-multiply(n ; integer);

```
var
    i, j, k; integer;
begin
    for i:=1 to n do
        for j:=1 to n do
            begin
                c[i, j]:=0;
                for k:=1 to n do
                    c[i, j]:=c[i, j]+a[i, k]*b[k, j]
                end;
            end;
        end;
    end;
```

(2) procedure add(n ; integer);

```
var
    i, j, k; integer;
    x, y; real
begin
    x:=0; y:=0
    for i:=1 to n do
        for j:=1 to i do
            for k:=1 to j do x:=x+y
        end;
    end;
```

(3) procedure while(n ; integer);

```
var
    i, j; integer;
begin
    i:=1; j:=1;
    while (i<=n and j<=n) do
```

```

        begin i:=i+1;j:=j+i; end;
    end;
(4) procedure until(n;integer);
    var
        i,j;integer;
    begin
        i:=1;
        repeat for j:=1 to n do i:=i+j;
        until i>=100+n
    end;
(5) function fact(n;integer);integer;
    begin
        if n<=1 then fact:=1
            else fact:=n * fact(n-1)
    end;

```

1.13 [05] 如果新计算机的速度是原计算机速度的 100 倍, 那么是否所有算法可能解决的问题长度(规模)都是原来的 100 倍?

1.14 [06] 按照阶由低到高的顺序排列下列时间复杂度: $O(2^n)$, $O(1)$, $O(n!)$, $O(n)$, $O(n^n)$, $O(n \log_2 n)$, $O(\log_2 n)$, $O(n^2)$, $O(n^{2/3})$, $O(n^{1/2})$, $O((2/3)^n)$, $O(n^3)$ 。

1.15 [10] 已知有两个算法实现同一功能, 时间复杂度分别为 $O(2^n)$ 和 $O(n^{10})$, 假设计算机可连续运行 10^7 秒, 每秒可执行基本操作 10^5 次, 可解问题的规模, 即 n 的范围各为多少? 用哪个算法较好? 为什么?

第二章 线性结构

本章重点:

1. 理解线性表的逻辑结构特性。
2. 熟练掌握线性表的顺序存储结构的描述方法,以及在该存储结构下的基本操作,如插入、删除等。
3. 熟练掌握线性表的链表存储结构的描述方法,灵活使用单链表、双链表、循环链表,学会在相应结构下实现线性表的各种运算。
4. 熟悉栈和队列这两种特殊线性结构的特性,并能在各种问题中灵活使用,熟练掌握栈列在顺序存储结构和链表存储结构下的基本运算。
5. 熟练掌握字符串的逻辑结构特性,熟悉字符串在顺序存储结构和链表存储结构下各种基本运算的实现方法,掌握 KMP 模式匹配算法。
6. 学会分析不同存储结构下各种算法的时间复杂度和空间复杂度,对于具体问题能够选择一种高效的(或较好的)算法去解决。
7. 理解递归概念,掌握如何使用栈结构来实现递归算法,能够将递归算法转化为非递归算法。

习题二 线性结构

2.1 [10]试比较顺序存储结构和链表存储结构有什么优缺点?

★2.2 [20]在顺序存储结构下实现线性表下列运算,并将它封装在单元中:

- (1) `crtsqlst(a)`。建立线性表 `a` 顺序存储结构;
- (2) `lensqlst(a)`。求线性表长度函数;
- (3) `locsqlist(a,x)`。定位函数,返回 `x` 在 `a` 中的位置;
- (4) `getsqlist(a,i,x)`。取线性表中第 `i` 个数据元素放在 `x` 中;
- (5) `inssqlist(a,i,x)`。在 `a` 中第 `i` 个数据元素前插入元素 `x`;
- (6) `delsqlist(a,i,x)`。删除 `a` 中第 `i` 个数据元素,被删元素放入 `x` 中;
- (7) `prnsqlist(a)`。打印 `a` 中所有数据元素;
- (8) `fullsqlist(a)`。如果线性表长度超过数组最大长度,返回值 `TURE`,否则返回值 `FALSE`;
- (9) `empsqlist(a)`。如果线性表为空,返回值 `TRUE`,否则返回 `FALSE`。

★2.3 [10]设两个递增有序线性表 `A` 和 `B`,均采用顺序存储结构,同一表中元素各不相同。试编写一个算法,将 `A` 和 `B` 合并为递增有序线性表 `C`(表 `C` 可以另辟空间,但要求表中元素值各不相同)。

2.4 [20]设有一个线性表采用顺序存储结构,表中元素值为正整数,元素个数为 `n`,试在 `O`

(n)时间内,将线性表分为两部分,其中在左半部分每个元素都小于(或大于)原表的第一个元素,而右半部分则相反。

★2.5 [25] 在带头结点单链表结构下实现线性表下列运算,并将它封装在单元中:

- (1) $\text{intlkst}(\text{head})$ 。初始化;
- (2) $\text{lenlkst}(\text{head})$ 。求单链表的长度;
- (3) $\text{getlkst}(\text{head}, i)$ 。函数,取单链表的第 i 个元素的结点指针;
- (4) $\text{inslkst}(\text{head}, i, b)$ 。在单链表的第 i 个元素之前插入元素 x ;
- (5) $\text{dllkst}(\text{head}, i)$ 。删除单链表的第 i 个结点;
- (6) $\text{crtlkst}(\text{head})$ 。对线性表建立单链表存储结构;
- (7) $\text{loclkst}(\text{head}, x)$ 。定位函数。确定元素在单链表中的位置,用指针表示;
- (8) $\text{prnlkst}(\text{head})$ 。打印单链表中元素;
- (9) $\text{emplkst}(\text{head})$ 。如果单链表为空,返回值 TURE,否则返回值 FALSE。

2.6 [10] 已知指针 ha 和 hb 分别指向两个单链表的头结点,头结点的数据域中存放链表的长度,试编写一个高效算法,将这两个链表连接在一起(即将一个表的头元素连在另一个表的尾元素之后),并分析算法的时间复杂度。

2.7 [15]] 已知指针 ha 指向某个单链表 LA 的头结点,试编写算法,完成下列运算: (1) 交换单链表中第 m 个和第 n 个结点的位置;
(2) 将单链表 LA 的内容复制到另一个单链表 LB 中。

2.8 [20] 设线性表 $A = (a_1, a_2, \dots, a_m)$, $B = (b_1, b_2, \dots, b_n)$, 试编写一个算法,将线性表 A, B 合并为线性表 C, 使

$$C = \begin{cases} (a_1, b_1, \dots, a_m, b_m, b_{m+1}, \dots, b_n) & \text{当 } m \leq n \text{ 时} \\ (a_1, b_1, \dots, a_n, b_n, a_{n+1}, \dots, a_m) & \text{当 } m > n \text{ 时} \end{cases}$$

要求 A, B 和 C 均以单链表为存储结构,且 C 表利用 A 和 B 中结点空间。这里 m 和 n 的值没有保存在头结点中。并分析算法时间复杂度。

★2.9 试编写算法,实现线性表的就地逆置,即在原表的存储空间内将线性表 $(a_1, a_2, \dots, a_n)$ 逆置为 $(a_n, a_{n-1}, \dots, a_2, a_1)$:

- (1) [10] 以一维数组作存储结构;
- (2) [25] 以单链表作存储结构。

★2.10 [30] 已知两个按元素值递增有序排列的线性表 A 和 B, 试编写算法,将这两个线性表合并为一个按元素值递减有序排列的线性表 C(允许值相同), 要求:

- (1) 采用顺序存储结构,可另外开辟空间存放 C;
- (2) 以单链表作存储结构,利用原表空间(即 A 和 B 的结点空间)存放 C。

2.11 [10] 假设有两个带头结点的循环单链表 A 和 B, 头结点的数据域中存放链表的长

度, ta 和 tb 分别指向 A 和 B 的最后一个元素, 试编写一个算法, 将 A 和 B 合并为带头结点的循环单链表 C。

2.12 [15] 设 L 指向循环单链表的头结点, 试编写一个算法, 将 L 分裂成两个循环链表 LA 和 LB , 分开后第 i 个结点以前所有结点构成 LA 表, 第 i 个结点以后的结点构成 LB 表(包括第 i 个结点)。

★2.13 [35] 约瑟夫问题可描述为: 编号为 $1, 2, \dots, n$ 的 n 个人按顺时针方向围坐一圈, 每人持有一个密码(正整数), 开始选一正整数 m , 从第一人开始按顺时针方向自 1 开始顺序报数, 报到 m 时此人出列, 将他的密码作为新的 m 值, 从他在顺时针方向的下一个人开始重新从 1 报数, 如此下去, 直至所有人全部出列。试编写算法, 打印出列顺序。要求:

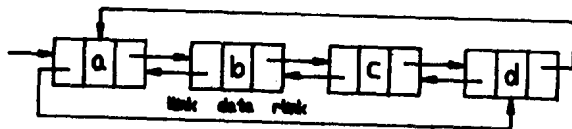
- (1) 采用顺序存储结构;
- (2) 采用循环单链表存储结构。

2.14 [10] 试编写一个算法, 将循环单链表改为循环双链表。设循环单链表中每个结点有一个域 pre , 但其值为 nil 。本算法要求使该指针指向其前驱结点。

★2.15 [25] 编写算法, 在循环双链表结构下实现线性表的下列运算, 并封装在单元中:

- (1) $crtdulkst(head, tail)$ 。输入元素值, 以此为数据域建立循环双链表存储结构;
- (2) $getdulkst(head, i)$ 。求线性表第 i 个数据元素的指针 p ;
- (3) $insdulkst(head, i, x)$ 。在第 i 个结点前插入元素值为 x 的新结点;
- (4) $deldulkst(head, i, p)$ 。删除第 i 个结点, 被删结点的指针为 p ;
- (5) $prndulkst(head)$ 。打印循环双链表。

2.16 [10] 设有如图所示循环双链表 $L = (a, b, c, d)$ 。试写出将该表转换为 $L = (b, a, c, d)$ 的简单操作:



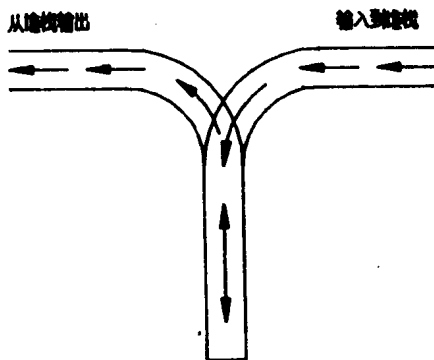
习题(2.16 图)

★2.17 [30] 已知由单链表表示的线性表中含有三类字符的数据元素(如数字字符、字母字符和其它字符), 试编写一个算法, 构造三个以循环单链表表示的线性表, 使每个表中只含同一类字符, 且利用原表的结点空间, 头结点可另辟空间。并分析算法的时间复杂度。

★2.18 [35] 试以循环单链表作稀疏多项式的存储结构, 写出两个多项式相加的算法, 并分析算法的时间复杂度。

2.19 [40] 试以循环单链表作稀疏多项式的存储结构,写出两个多项式相乘的算法,并分析算法的时间复杂度。

2.20 [05] 想象六辆列车位于图中堆栈的输入一边,列车编号为 123456,按此顺序开进堆栈,且可在任意时刻开走,则能否得到 325641 出站序列? 能否得到 154623 出站序列? 在可能的情况下,说明如何实现。



习题 2.20 图

★2.21 [30] 假设以 S 和 X 分别表示入栈和出栈操作,则初态和终态均为栈空的入栈和出栈的操作序列可以表示为仅由 S 和 X 组成的序列。称 S 和 X 的一个序列为合法的,如果它不指定不可能实现的操作(如 SXXSSX 为非法序列,SSXX 为合法序列),试描述一个规则,使用它能容易地区别合法与非法序列。并证明:两个不同的合法序列(对同一输入序列)不可能得到相同的输出序列。

2.22 [20] 在习题 2.20 图所示的堆栈式站台中,若输入序列为 87654321,则可能出现的输出序列有多少种?

2.23 [35] 证明:有可能利用一个栈从 $1, 2, \dots, n$ 得到排列 $p_1, p_2, \dots, p_n$, 其充分必要条件是:不存在下标 $i < j < k$, 使得 $p_i < p_k < p_j$ 。

2.24 [05] 下题中 s 表示栈名, a、b 为栈中元素,试说明下列运算的结果:

- (1) POP(PUSH(s, a));
- (2) PUSH(s, POP(s));
- (3) PUSH(s, POP(PUSH(s, b)))。

★2.25 [20] 试编写算法,在顺序存储结构下实现堆栈的下列运算,并将它封装在单元中:

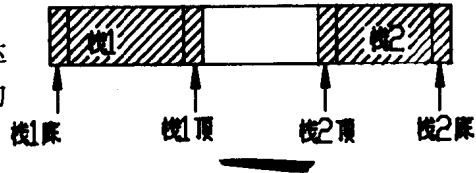
- (1) initstk(s)。初始化操作,建立一个空栈 s;
- (2) emptystk(s)。判断栈是否为空;
- (3) pushstk(s, x)。如果栈 s 不满,在栈顶插入 x;
- (4) popstk(s)。如果栈 s 不空,删除栈顶元素,并返回该元素的值;
- (5) getstk(s)。如果栈 s 不空,返回栈顶元素;

- (6) clearstk(s)。将栈清为空栈;
 (7) lenstk(s)。求当前栈中元素个数。

2.26 [20] 如图所示两个迎面增长的栈 S1 和 S2 共享空间 V(1:n), 试编写算法, 对其中任一栈进行 push, pop 运算, 要求只有整个空间 V(1:n) 占满时才产生上溢。

★2.27 [20] 试编写算法, 在单链表存储结构下实现习题 2.25 所列堆栈的(1)——(7)种运算, 并将它封装在单元中。

2.28 [10] 采用算符优先算法对中缀算术表达式求值, 画出对下式求值时操作符栈和运算符栈的变化过程: $x + y * (u - v / w)$ 。



习题 2.26 图

★2.29 [30] 编写一个算法, 对后缀表达式求值。假设表达式由单字母变量和双目四则运算算符组成。

2.30 [25] Fibonacci 序列可递归定义为

$$\text{Fib}(n) = \begin{cases} n, & n=0, 1 \\ \text{Fib}(n-2) + \text{Fib}(n-1), & n \geq 2 \end{cases}$$

试编写一个递归算法, 并利用栈来模拟递归执行过程, 消除递归。

★2.31 [40] Hanoi 塔问题: 有三个轴 A, B, C, n 个大小不同的圆盘插在 A 轴上, 依小到大连号为 1, 2, ..., n, 现要求将 A 轴上 n 个圆盘移至 C 轴上并按同样顺序排列, 圆盘移动时规定: 每次只能移动任一轴上最顶部的一个圆盘, 可以随意使用这三个轴, 但要保证任何时刻每个轴上的圆盘都满足是一个塔形(上小下大), 试编写一个算法, 解 Hanoi 塔问题, 并对 n=4 的情形, 打印移动步骤。

★2.32 [45] 背包问题: 设有一个背包可装入物品总重量为 T, 现有 n 件物品, 重量分别为 $W_1, W_2, \dots, W_n$, 问能否从这 n 件物品中选择若干件放入背包, 使放入的重量之和恰好为 T, 若能找到满足上述条件的一组解, 则称此问题有解, 否则称为无解。试编写一个算法, 求解背包问题。并对 $T=20, n=5$, 重量分别为 {1, 8, 7, 3, 11}, 打印其结果。

2.33 [20] 当输入元素序列为 54321 时, 试求双向队列的输出序列共有多少种?

2.34 [20] 利用两个栈 S1, S2 模拟一个队列时, 如何用栈的运算实现下列运算? enqueue(入队), dlqueue(出队), empty(判空), 试编写算法。

★2.35 [20] 假设以顺序结构 $\text{sequeq}(0..m-1)$ 存储循环队列的元素, 试完成队列的下列运

算,并将它封装在单元中:

- (1) initcycq. 建立一个空队列;
- (2) encycq. 如果队列不满,在队列中插入一个新元素;
- (3) dlcyq. 如果队列不空,在队头删除一个元素;
- (4) getcycq. 如果队列不空,取队头元素并返回;
- (5) fullcycq. 判断队列是否为满;
- (6) emptycycq. 如果队列不空,返回值为 TURE,否则为 FALSE;
- (7) clearcycq. 清队列为空;
- (8) lencycq. 求队列长度;
- (9) prncycq. 打印循环队列。

★2.36 [20] 试在链表存储结构下完成队列的下列运算,并将它封装在单元中:

iniquueue, enqueue, dlqueue, getqueue, empty, length。

2.37 [20] 一个双向队列是允许在表的两端插入和删除的队列。试编写一个算法,在链表存储结构下实现双向队列的入队和出队操作。

2.38 [30] 以双排队代替堆栈,考虑下列问题:

(1) 试求 1234 的一个排列,它能够通过利用一个输入受限的双排队得到,但不能通过输出受限的双排队得到;

(2) 试求 1234 的一个排列,它能够通过利用一个输出受限的双排队得到,但不能通过输入受限的双排队得到;

(3) 试求 1234 的一个排列,它不论通过一个输入受限的双排队,还是通过输出受限的双排队,都不能得到。

2.39 [35] 通过使用一般的双排队,可以得到多少 n 个元素的排列?

2.40 [02] 设 $a = ''$, $b = \text{'old friend'}$, $c = \text{'good'}$, $d = \text{'new'}$, 请完成下列运算:

- (1) LEN(a), LEN(b);
- (2) SUB(b , 4, 3);
- (3) INDEX(c , 'o'), INDEX(b , 'd');
- (4) REPLACE(b , 'old', d);
- (5) CONCAT(c , CONCAT(a , b))。

2.41 [10] 设 $A = \text{'(xyz) + *'}$, $B = \text{'(x + z) * y'}$ 。试利用串的基本操作,将 A 转化为 B 。

★2.42 [25] 假设以顺序存储结构存放字符串,试编写算法,实现下列基本操作,并将它封装在单元中:

- (1) creatstring(s)。输入并建立顺序存储的字符串;
- (2) concat(x , s , t)。连接两个字符串 s 和 t , 放入 x 中;
- (3) substring(sub , s , start, len)。求 s 串从 start 开始长度为 len 的子串放入 sub ;
- (4) index(s , t)。定位函数, s 为主串, t 为模式串;
- (5) instring(s , i , t)。在串 s 的第 i 个字符前插入 t ;

- (6) `delstring(x,s,i,len)`。从串 s 中删除第 i 个字符起长度为 len 的子串送到 x 中;
- (7) `equal(s,t)`。判断两串是否相等;
- (8) `prnstr(s)`。打印字符串。

★2.43 [25] 假设以单链表存储结构存放字符串,且每个结点只存一个字符。试编写算法,实现如下基本操作,并将它封装在单元中。

- (1) `asnlkstr(hs,ht)`。将字符串 ht 的值送给 hs ;
- (2) `conlkstr(hs,ht)`。连接两个字符串 hs 和 ht ;
- (3) `sublkstr(hsub,hs,start,len)`。求 hs 串从 $start$ 开始长度为 len 的子串放入 $hsub$;
- (4) `inslkstr(hs,ht,i)`。将字符串 ht 链在字符串 hs 的第 i 个字符之前;
- (5) `dellkstr(hs,i,len)`。删除 hs 字符串中从第 i 个字符开始长度为 len 的字符串;
- (6) `crtlkstr(hs)`。输入字符串,建立链表存储结构;
- (7) `indxlkstr(hs,ht)`。定位函数,即模式匹配;
- (8) `prnlkstr(hs)`。打印字符串;
- (9) `eqlkstr(hs,ht)`。判断两个字符串是否相等。

2.44 [10] 在链表存储结构中,每个结点存放 4 个字符。试编写一个算法,删除串 S 中第 i 个字符开始的 len 个字符,并计算该算法的时间复杂度。

★2.45 [10] 令 $t1='abcabaa'$, $t2='aabbaab'$, 试利用 KMP 算法和改进算法分别求它们的 NEXT 函数值和 NEXTVAL 函数值。

2.46 [15] 对 $S='baabababbaabba'$, $T='aabb'$, 按 KMP 算法进行模式匹配,画出计算过程图。

★2.47 [30] 假设以无表头结点的单链表存储串值,且每个结点中含有 4 个字符,已知串长为 n ,试编写一个算法,判别该串是否具有对称性,要求算法时间复杂度为 $O(n)$ 。

2.48 [20] 如果 $X=(x_1, \dots, x_m)$, $Y=(y_1, \dots, y_n)$ 是由字母组成的字符串。试编写一个算法,按字典序比较 X 串和 Y 串,按 $X < Y$, $X = Y$, $X > Y$ 三种情况,分别返回 $-1, 0, +1$ 值。

软件水平考试题选

★2.49 请从下列关于数据结构的叙述中选出正确的叙述:

1. 数据元素是数据的最小单位。
2. 数据结构是带有结构的数据元素的集合。
3. 数据结构、数据元素、数据项在计算机中的映象(或表示)分别称为存储结构、结点、数据域。
4. 线性表的线性存储结构优于链式存储结构。
5. 二叉树的第 i 层上有 2^{i-1} 个结点,深度为 k 的二叉树上有 $2^k - 1$ 个结点。
6. 子串定位函数的时间复杂度在最坏情况下为 $O(n * m)$,是不能实际使用的。