

北京工业大学 1997 年数据结构试题

一. (共 20 分)

1. (8 分) 试用类 Pascal 语言编写过程 PROC JOIN (VAR LA: LINK; LB: LINK) 实现连接线性表 L_a 和 L_b (L_b 在后) 的算法, 要求其时间复杂度为 O(1), 占用辅助空间尽量小。描述所用结构。
2. (12 分) 顺序结构线性表 LA 与 LB 的元素按非递减有序, 线性表空间足够大。试用类 PASCAL 语言给出一种高效算法, 将 LB 中元素合到 LA 中, 使新的 LA 的元素仍保持非递减有序。高效制最大限度的避免移动元素。

二. (31 分)

1. (10 分) 设二叉数采用二叉链表作为存储结构。设栈已经定义: INITS (S), EMPTY (S) PUSH (S, P), POP (S), TOP (S) 分别为栈初始化, 判栈空, 入栈, 出栈, 看栈顶等操作。
2. (10 分) 下面的类 PASCAL 语言递归算法的功能是判断一棵二叉数 (采用二叉数表存贮结构) 是否为完全二叉数。请把空缺的两部分补写完整。(提示: 利用完全二叉数结点序号性质)

TYPE

link=^node;
node=RECORD
 key:keytype;
 l,r:link;

END;

VAR all:boolean; n:integer; root:link;

FUNC num(t:link):integer;

BEGIN

(1)

END;

PROC chk(t:link;m{t 所指结点应有序号}:integer);

BEGIN

(2)

END;

BEGIN {建二叉数, 其根由 root 指出 }

n:=num(root); {求结点数} all:=true;

chk(root,1);

IF all

THEN writeln ('该树为完全二叉数!')

ELSE writeln ('该树非完全二叉数!')

END;

3. (5 分) 用关键字 1, 2, 3, 4 的四个结点 (1) 能够造出几种不同的二叉排列树? 其中 (2) 最优查找树有几种? (3) AVL 树有几种? (4) 完全二叉树有几种? 试画出这些二叉排序树。

4. (6 分) 已知某系统在通讯联络中只可能出现八种字符 d₁——d₈, 出现频率分别为 0.03, 0.04, 0.05, 0.06, 0.15, 0.19, 0.20, 0.28。试为止设计哈夫曼编码 (要求: 按给出的数

据和哈夫曼树建树算法，用示意图分别给出哈夫曼树的初态，终态，哈夫曼编码表，并画出这棵哈夫曼树）。

三、(10 分) 以三元组表存贮的稀疏矩阵 A, B 非零元个数分别为 m 和 n。试用类 PASCAL 语言编写时间复杂度为 $O(m+n)$ 的算法将矩阵 B 加到矩阵 A 上去。A 的空间足够大，不另加辅助空间。要求描述所用结构。

四、(15 分) 对于顶点 A——M 和输入的弧序列：AB, AC, AF, AL, BM, LJ, LM, MJ, DE, GH, GI, GK, HK, 按照弧结点在链头插入的规则构造出有向图的十字链表存贮结构图；按此结构来给出广度优先遍历顶点序列及相应的生成森林，画出图和森林的示意图。

五、(共 24 分)

1、(8 分) 用依次输入的关键字 13, 20, 41, 19, 17, 5, 1, 7 和 6 键一棵 AVL 树，画出建该树的变化过程示意图（每引起一次调整至少有一张图并写出新插入的关键字）。

2、(16 分) 将如下的堆排序算法补写完整。说明如下：

TYPE heaptype=ARRAY[1..NO]OF integer;

过程 heapsort 的功能是将数组 h 中的前 n 个记录按关键字递减的次序排序。Heapsort 调用过程 sift 时的参数 h, k, l 有如下定义：以 $h[k+1], h[k+2], \dots, h[r]$ 为根的子树已经是堆；执行 sift 后，以 $h[k], h[k+1], \dots, h[r]$ 为根的子树都成为堆。

PROC sift (VAR h: heaptype; k, r: integer);

VAR i, j, x: integer; finish: boolean;

BEGIN i:=k; x:=h[i]; j:=2*i;

(1)

WHILE (j<=r)AND NOT finishDO

[if (J<R) AND (H[J]>h[j+1]) THEN j:=j+1;

IF X>H[J] THEN

[

(2)

]

ELSE finish:=true;

(3)

]

END;

PROC heapsort(VAR h:heaptype;n:integer);

VAR k,r,i,j:integer;

BEGIN

FOR k:=n DIV 2 DOWNT0 1 DO

Sift((4))

FOR r:=n DOWNT0 2 DO

[x:=h[1]; h[1]:=h[r]; h[r]:=x;

(5)

]

END.