

中国科学院软件研究所

一九九七年招收硕士学位研究生入学考试试题

试题名称：软件基础

第一部分：数据结构和程序设计

一. 是非题 (正确填√, 错误填×) (1分×10)

- () 1. 稀疏矩阵压缩存储后, 必会失去随机存取功能。
- () 2. 完全二叉树的某结点若无左孩子, 则它必是叶结点。
- () 3. 由一棵二叉树的前序序列和后序序列可以唯一确定它。
- () 4. 在 n 个结点的无向图中, 若边数 $> n-1$, 则该图必是连通图。
- () 5. 若一个有向图的邻接矩阵中对角线以下元素均为零, 则该图的拓扑有序序列必定存在。
- () 6. 用向量和单链表表示的有序表均可使用折半查找方法来提高查找速度。
- () 7. 在任意一棵非空二叉排序树, 删除某结点后又将其插入, 则所得二叉排序树与删除前原二叉排序树相同。
- () 8. 若一个广义表的表头为空表, 则此广义表亦为空表。
- () 9. 二叉树是度数为二的有序树。
- () 10. 对磁带机而言, ISAM 是一种方便的文件组织方法。

二. 程序阅读题 (共 18 分)

1. 设 m, n 均为自然数, m 可表示为一些不超过 n 的自然数之和, $f(m,n)$ 为这种表示方式的数目。例 $f(5,3)=5$, 有 5 种表示方式: $3+2, 3+1+1, 2+2+1, 2+1+1+1, 1+1+1+1+1$ 。

① 以下是该函数的程序段, 请将未完成部分填入, 使之完整。

```
int f(m,n)
    int m,n;
    {
        if (m==1)
            return _ _ _ _ _ ;
        if (n==1) {
            return _ _ _ _ _ ;
        }
        if (m<n) {
            return f(m,m);
        }
        if (m==n) {
            return 1+ _ _ _ _ _ ;
        }
        return f(m,n-1)+f(m-n, _ _ _ _ _ );
    }
```

② 执行程序, $f(6,4)=$ _ _ _ _ _ ;

2. 设输入为整数数组 $A[1..n]$, 其中 $1 \leq A[i] \leq k$ ($1 \leq i \leq n$); 输出数组为 $B[1..n]$; $C[1..k]$ 是临时工作空间; 阅读下述算法后, 回答下列问题:

```

Proc Demo(A,B,k) {
    (1) for i:=1 to k do C[i]:=0;
    (2) for j:=1 to n do C[A[j]]:=C[A[j]]+1;
    (3) for i:=2 to k do C[i]:=C[i]+C[i-1];
    (4) for j:=n downto 1 do {
    (5)     B[C[A[j]]]:=A[j];
    (6)     C[A[j]]:=C[A[j]]-1;
    }
}
    
```

- (a) 当标号(2)行的循环执行完后, $C[i]$ ($1 \leq i \leq n$) 的值有何意义?
- (b) 当标号(3)行的循环执行完后, $C[i]$ ($1 \leq i \leq n$) 的值有何意义?
- (c) 算法执行后, B 的内容有何特点?
- (d) 当 $k=O(n)$ 时, 算法的时间复杂度是多少?

三. (10分) 设大根堆 (即堆顶元素最大) 的存储结构为:

```

type HEAP=record
    keys: array[1..max] of keytype; /*存放堆元素的空间*/
    size: int; /*当前堆中已有的元素个数*/
end;
    
```

写一算法 `HeapInsert(var heap:HEAP;key:keytype)` 将关键字 `key` 插入到堆 `heap` 中且不破坏堆性质。(提示: 先将 `key` 插入到数组的 `size+1` 位置上, 然后自底向上调整)。

四. (12分) 设二叉排序树的存储结构为:

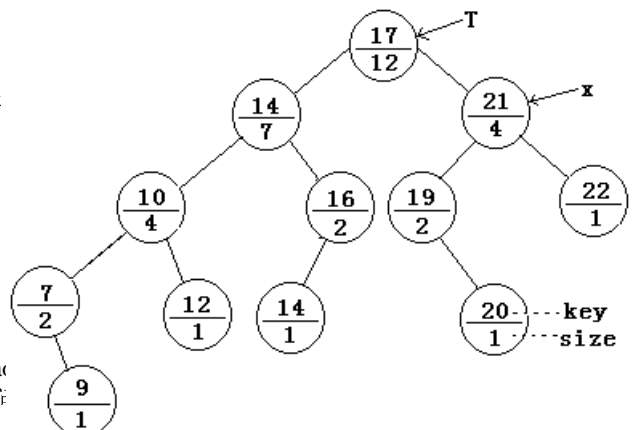
```

type TREE= ↑ node;
node = record
    key : keytype;
    size : int;
    lchild, rchild, parents: TREE;
end;
    
```

一个结点 x 的 `size` 域的值是以该结点为根的子树中结点的总数 (包括 x 本身)。例如, 下图中 x 所指结点的 `size` 值为 4。设树高为 h , 试写一时间为 $O(h)$ 的算法 `Rank (T: TREE; x: ↑ node)` 返回 x 所指结点在二叉排序树 T 的中序序列里的排列序号, 即:

求 x 结点是根为 T 的二叉排序树中第几个最小元素。例如, 下图 x 所指结点是树 T 中第 11 个最小元素。

(提示: 你可利用 `size` 值和双亲指针 `parents`)。



第二部分：操作系统和编译技术

五. 填空题 (1分×9)

1. 实现虚拟存储器，应有以下几种物质基础来支持：① _____
 ② _____ ③ _____。
2. 页面淘汰算法选得不好会引起 _____ 现象。
3. 从资源管理的角度看，I/O 设备分为 _____ 设备、_____ 设备和 _____ 设备三类。
4. 引入缓冲的原因在于：_____ 和 _____。

六. 简答题 (共 16 分)

1. 一个分层结构操作系统有下列部分组成：裸机、用户、CPU 调度和 PV 操作、文件管理、作业管理、内存管理、设备管理、命令管理等。试按层次结构的原则从内到外将各部分重新排列。
2. 产生死锁的必要条件是什么？解决死锁问题常用哪几种措施？
3. UNIX 进程映像由哪几部分组成？简述各部分的内容。

七. (10 分) 为正规式 $(a|b)^*a(a|b)$ 构造一个确定的有限自动机。

八. (15 分) 试画出如下中间代码序列的程序流图，并求出：

- (1) 各结点的必经结点集合 $D(n)$;
- (2) 流图中的回边与循环。

```

J:=0;
L1: I:=0;
    if I<8 goto L3;
L2: A:=B+C;
    B:=D*C;
L3: if B=0 goto L4;
    write B;
    goto L5;
L4: I:=I+1;
    if I<8 goto L2;
L5: J:=J+1;
    if J<=3 goto L1;
    HALT
    
```

中国科学院软件研究所

一九九七年招收硕士学位研究生入学考试试题答案

试题名称：软件基础

一. 是非题 (1分×10)

1. √ 2. √ 3. × 4. × 5. √ 6. × 7. × 8. × 9. × 10. ×

二. 程序阅读题 (共 18 分)

1. (2分×5)

① 1, 1, f(m, n-1), n② 9 (nine)

2. (2分×4)

(a) C[i]表示数组 A 中值等于 i 的元素的个数 ($1 \leq i \leq n$);(b) C[i]表示数组 A 中值小于等于 i 的元素的个数 ($1 \leq i \leq n$);

(c) B[1..n]有序

(d) O(n)

注：这是一个计数排序算法。

三. (10分)

```
proc HeapInsert(var heap:HEAP; key:keystype)
{
  if heap.size  $\geq$  max
  then error("overflow")
  else
  {
    heap.size := heap.size + 1;
    i := heap.size; //新结点作为树叶先插入 i 位置
    while (i > 1) and (heap.keys[parents(i)] < key) do
    { //向上调整最多至根
      heap.keys[i] := heap.keys[parents(i)];
      i := parents(i); //相当于新结点往上调整至双亲
    } //end of while
    heap.keys[i] := key
  } //end of if
} //end of proc
```

这里 parents(i)是一函数，返回值为 $\lfloor i/2 \rfloor$ 。

此题要点如题中的提示，必须从 size+1 位置往上调整使其满足堆性质；另一要点是将数组 keys 看作是完全二叉树的顺序存储结构，若从上往下调整，难以保证筛下的元素恰好是数在 size+1 的位置。

四. (12分)

```

func Rank (T: TREE; x: ↑node): int
{ //可设 x≠nil
  if x↑.lchild = nil
  then r := 1
  else r := x↑.lchild↑.size + 1;
  y :=x;
  while y ≠ T do
  {
    //循环不变量:r 表示以 y 为根的子树中, 结点 x↑ 的 Rank 值
    //当 y 等于根 T 时, 即为所求
    if y = y↑.parents↑.rchild
    then
      {
        //当 y↑ 是其双亲的右子时, y↑ 的双亲结点的左子树中所有结点
        //y↑ 的双亲本身均应排在 y↑ 的前面(指中根序列)
        if y↑.parents↑.lchild = nil
        then r := r + 1;
        else r := r + y↑.parents↑.lchild↑.size + 1;
      }
      y := y↑.parents;
    }//end of while
  }return r;
} //end of func.
    
```

此题要点: 先找 $x \uparrow$ 在以 $x \uparrow$ 为根的子树中的排列序号 r , 因为 $x \uparrow$ 的左子树中所有结点在中根序列中均处于 $x \uparrow$ 之前, 所以有:

$$r := x \uparrow \text{左子树的大小 (size)} + 1;$$

这里加 1 是包括 $x \uparrow$ 本身。当 $x \uparrow$ 左子树为空时, 相当于左子树 size 为 0。

注意: $x \uparrow$ 在以 $x \uparrow$ 为根的子树中的 r 值并非它在整个树 T 中的排列序号, 所以要从 $x \uparrow$ 上溯到其双亲, 当 $x \uparrow$ 是其双亲的左子树时, r 值不变; 当 $x \uparrow$ 是其双亲的右子树时, 必须加上双亲结点本身及其左子树中所有结点数作为新的 r 值。此过程最多上溯至根。所以执行时间不超过树的高度。

五. (1分×9)

- ①相当容量的辅存; ②一定容量的主存; ③地址变换机构。
- 抖动。
- 独占、共享、虚拟。
- 提高硬件的并行操作程度、减少对 CPU 的中断次数。

六. (共 16 分, 第一题 4 分, 其余二题各 6 分)

- 裸机、CPU 调度和 PV 操作、内存管理、作业管理、设备管理、文件管理、命令管理、用户。
- 产生死锁的必要条件有: ①独占条件 (互斥条件); ②请求和保持条件;

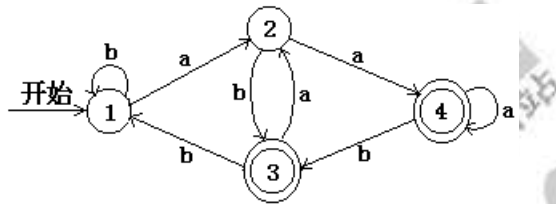
③不剥夺条件；④环路条件（循环等待条件）。

解决死锁问题常用的方法有：①死锁的检测和解除；②死锁的预防。

3. UNIX 进程映像由 PROC 结构、正文段、数据段三部分组成。

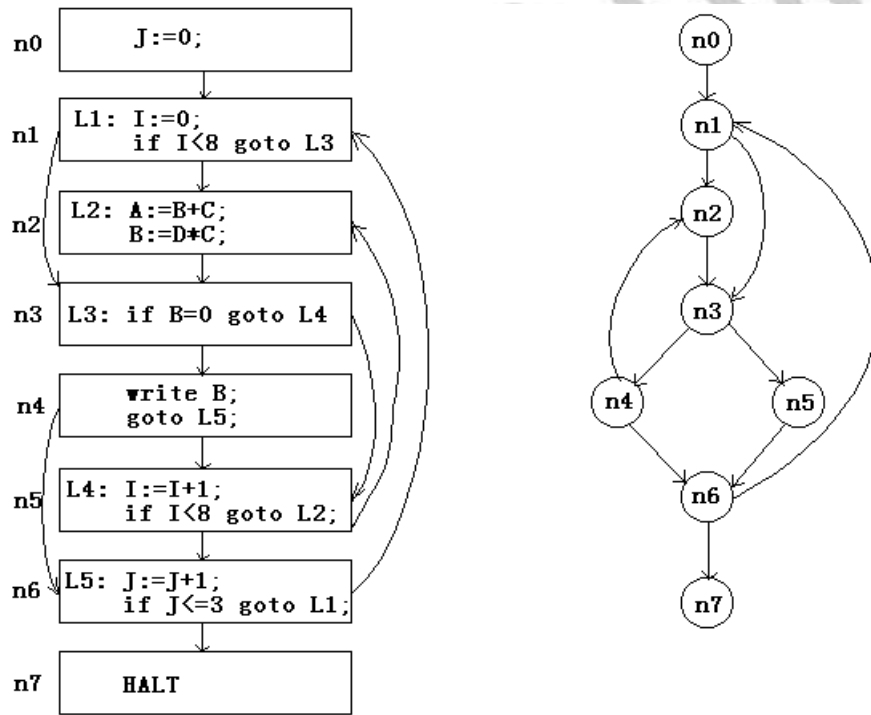
- PROC 结构为进程控制块 PCB 的一部分，它含有进程最常用的信息：进程状态、进程优先级、进程特征、地址和大小信息等。
- 正文段内容为可重入的程序代码，可以为若干个进程共享。
- 数据段由用户栈区、用户数据区、系统数据区三部分组成。

七. (10分)



八. (15分)

程序的流程为下，方框表示基本块。



(1) 各结点的必经结点集合分别为：

- $D(n0) = \{n0\}$
- $D(n1) = \{n0, n1\}$
- $D(n2) = \{n0, n1, n2\}$
- $D(n3) = \{n0, n1, n3\}$
- $D(n4) = \{n0, n1, n3, n4\}$
- $D(n5) = \{n0, n1, n3, n5\}$
- $D(n6) = \{n0, n1, n3, n6\}$
- $D(n7) = \{n0, n1, n3, n6, n7\}$

(2) 回边有 $n6 \rightarrow n1$ 一条。

该回边所表示的循环为 $\{n_1, n_2, n_3, n_4, n_5, n_6\}$ 。入口为 n_1 ，出口为 n_6 。

