

## 电子科技大学

## 2001 年攻读硕士学位研究生入学试题

## 科目名称: 软件基础

注: 一、二、三、四题为数据结构部分 (共 50 分), 其它为操作系统部分 (共 50 分)

## 一、单项选择题 (每题 1 分, 共 8 分)

1. 若表 R 在排序前已经按关键字值递增排列, 则 (1) 算法的比较次数最少。

(1) 直接插入排序

(2) 快速排序

(3) 归并排序

(4) 选择排序

2. 链表不具备的特点是 (1)。

(1) 可随机访问任一元素

(2) 插入删除不需要移动元素

(3) 不必事先预分存储空间

(4) 所需空间与线性表长度成正比

3. 在有  $n$  个叶结点的哈夫曼树中, 其结点总数为 (4)。

(1) 不确定

(2)  $2n$ (3)  $2n + 1$ (4)  $2n - 1$ 

4. 若线性表最常用的操作是存取第  $i$  个元素及其前趋的值, 则采用 (4) 存储方式节省时间。

(1) 单链表

(2) 双链表

(3) 单循环链表

(4) 顺序表

5. 若某链表最常用的操作是在最后一个元素之后插入一个结点和删除最后一个结点, 则采用 (2) (4) 存储方式节省时间。

(1) 单链表

(2) 双链表

(3) 单循环链表

(4) 带头结点的双循环链表

6. 判断线索二叉树中某结点  $P \uparrow$  有左孩子的条件是 (3)。

(1)  $P \neq \text{nil}$ (2)  $P \uparrow \cdot \text{lchild} \neq \text{nil}$ (3)  $P \uparrow \cdot \text{ltag} \rightarrow \text{ltag}$ (4)  $P \uparrow \cdot \text{ltag} = 1$ 

7. 已知二叉树中叶结点数为 50, 仅有一个孩子的结点数为 30, 则总结点数为 (2)。

(1) 81

(2) 129

 $n_0 = n_2$



2. 修改下面层次遍历二叉树算法，使其能判断该二叉树是否为完全二叉树。(8分)

PROC level (bt: bitreptr);

IF bt  $\diamond$  nil THEN

[ INIQUEUE(Q): ENQUEUE(Q, bt); {初始化队列 Q, 并将根入队列}

WHILE NOT EMPTY(Q) DO {队列非空进入循环}

[ p: =DEQUEUE(Q); {出队列}

visit(p  $\uparrow$ .data);

IF p  $\uparrow$ .lchild  $\diamond$  nil THEN ENQUEUE(Q, p  $\uparrow$ .lchild);

IF p  $\uparrow$ .rchild  $\diamond$  nil THEN ENQUEUE(Q, p  $\uparrow$ .rchild);

]

]

ENDP; { level }

参考算法:

▲ **算法思想:** 完全二叉树中，若某结点无左孩子，也不能有右孩子；

若某结点缺少孩子，其后的所有结点就不能再有孩子。

因此，可以使用一个标志 tag 和两个计数变量 i 和 j，每访问一个结点，i 加 1，

若某结点缺少孩子时，标志置为 false；计数变量 j 停止计数。

最后，当 i=j 时，为完全二叉树，否则为非完全二叉树。

PROC level (bt: bitreptr);

i: =0; j: =0; tag: =true; {初始化}

IF bt  $\diamond$  nil THEN

[ INIQUEUE(Q): ENQUEUE(Q, bt); {初始化队列 Q, 并将根入队列}

WHILE NOT EMPTY(Q) DO {队列非空进入循环}

[ p: =DEQUEUE(Q); {出队列}

i: =i+1; IF tag THEN j: =j+1;

IF p  $\uparrow$ .lchild  $\diamond$  nil THEN [ENQUEUE(Q, p  $\uparrow$ .lchild), IF tag THEN j:=j+1;]

ELSE tag: =false;

IF p  $\uparrow$ .rchild  $\diamond$  nil THEN [ENQUEUE(Q, p  $\uparrow$ .rchild), IF tag THEN j:=j+1;]

ELSE tag: =false;

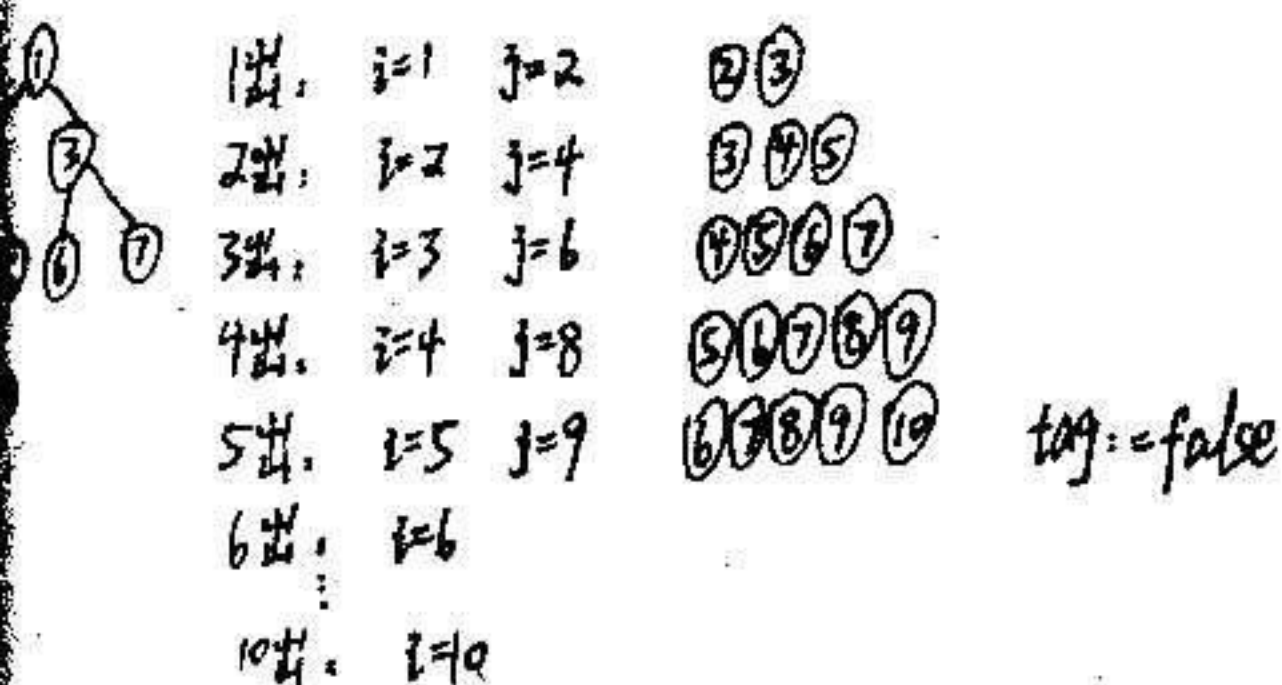
]

IF i > j THEN j:=j+1;

IF i=j THEN write('是完全二叉树')

ELSE write('不是完全二叉树')

ENDP; { level }





# 操作系统的参考答案

五. 单项选择题: (每小题 1 分, 共 10 分)

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

六. 填空题 (每小题 1 分, 共 9 分)

1. 512 字节    2. 根, 当前工作    3. 物理    4. 原子操作  
5. bernstein 条件    6. 竞争资源, 进程推进顺序不当  
7. 共享存储系统, 消息系统, 管道通信  
8. 脱机作业控制, 联机作业控制  
9. 字节多路通道, 数组选择通道, 数组多路通道

七. 判断题 (每小题 1 分, 共 10 分)

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

八. 问答题 (每小题 7 分, 共 21 分)

1. 答: 其它用户的平均等待时间:  $\frac{3}{250} \frac{250}{5}$

2. 答: 缺页次数: 6 次。缺率=0.3 [10分]

3. 答: CPU 的平均有效访问存储器时间为: 52

2. (LRU算法):

1 8 1 7 8 2 7 2 1 8 3 8 2 1 3 1 7 1 3 7





(3) 110

(4) 130

8. 将线性表的数据元素进行扩充, 允许是带结构的线性表是 (3)。

(1) 串  $\rightarrow$  无限制(2) 树  $\rightarrow$  前驱, 后继个数扩展(3) 广义表  $\rightarrow$  无限制(4) 栈  $\rightarrow$  操作限制

二、判断题: 正确用  $\checkmark$  表示, 错误用  $\times$  表示。(每题 1 分, 共 5 分)

1. ( $\times$ ) 在顺序表中取出第  $i$  个元素所花费的时间与  $i$  成正比。

2. ( $\checkmark$ ) 一个有向图的邻接表和逆邻接表中的结点个数一定相等。

3. ( $\checkmark$ ) 直接选择排序算法的时间复杂度为  $O(n^2)$ , 不受数据初始排列的影响。

4. ( $\times$ ) 由于希尔排序的最后一趟与直接插入排序过程相同, 因此前者一定比后者花费的时间更多。

5. ( $\times$ ) 广义表的长度是指广义表中的原子个数。

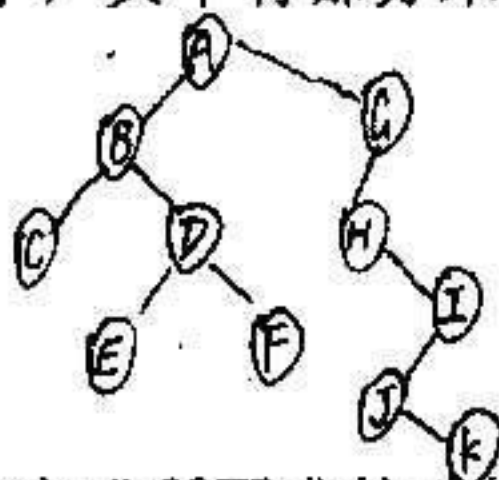
三、简答题 (每题 5 分, 共 20 分)

1. 一棵二叉树的先序、中序和后序序列如下, 其中有部分未标出, 试构造出该二叉树。

先序序列为: A B C D E F G H I J K

中序序列为: C B E D F A H J K I G

后序序列为: C E F D B K J I H G A



2. 将下面算法划线处用具体语句表示, 使其完成所要求的功能。

```
PROC ins_linklist (la : linkisttp; i: integer; b:
elemtp );
```

{la 为带头结点单链表的头指针, 在  $i$  之前插入数据元素  $b$ }

p: = la  $\uparrow$ .next; j: = 0;

WHILE ( 循环控制条件 ) DO

[ 循环体部分 ];

IF ( 判定条件 )

THEN ERROR ( ' i 非法' )

ELSE [ new(s); s  $\uparrow$ .data: =b;

插入链表部分

]

ENDP; { ins\_linklist }

p < > NIL AND j < i-1

p := p  $\uparrow$ .next; j := j+1

p = NIL OR ~~not~~ j > i-1

s  $\uparrow$ .next := p  $\uparrow$ .next;

p  $\uparrow$ .next := s

3. 对有向图和无向图, 分别描述如何判定图中是否存在回路。[10分]



4. 试设计一种链队列的存储结构, 要求只用一个指针, 并且使其插入和删除的时间复杂度均为  $O(1)$ 。试画出存储结构, 并指出队首和对尾。

#### 四、算法题 (共 17 分)

1. 修改下面中序遍历二叉树算法, 使其能判定所输出的序列是否有序。(9 分) [即修改为 BST 的判定算法]

```
PROC inorder (bt: bitreptr);
```

```
{bt 为二叉树根结点指针}
```

```
IF bt<>nil THEN
```

```
[ inorder(bt↑.lchild);
```

```
visit(bt↑.data);
```

```
inorder(bt↑.rchild)
```

```
]
```

```
ENDP; { inorder }
```

2. 修改下面层次遍历二叉树算法, 使其能判断该二叉树是否为完全二叉树。(8 分)

```
PROC level (bt: bitreptr);
```

```
IF bt<>nil THEN
```

```
[ INIQUEUE(Q); ENQUEUE(Q, bt); {初始化队列 Q, 并将根入队列}
```

```
WHILE NOT EMPTY(Q) DO {队列非空进入循环}
```

```
[ p:=DEQUEUE(Q); {出队列}
```

```
visit(p↑.data);
```

```
IF p↑.lchild<>nil THEN ENQUEUE(Q, p↑.lchild);
```

```
IF p↑.rchild<>nil THEN ENQUEUE(Q, p↑.rchild);
```

```
]
```

```
]
```

```
ENDP; { level }
```

五、单项选择题 (请选出正确的编号填入括号中, 每小题 1 分, 共 10 分)



1. 进程被阻塞以后, 代表进程在阻塞队列的是它的 ( ② )  
 ①文件控制块      ②进程控制块      ③作业控制块  
 ④设备控制块
2. 在以下哪种状态下, 作业已获得虚处理机。( ① )  
 ①提交状态      ②运行状态      ③输入状态  
 ④输出状态
3. 进程调度程序的主要功能是 ( ③ )  
 ①在作业和作业之间切换 CPU      ②防止死锁  
 ③在进程和进程之间切换 CPU      ④实现进程之间的同步与互斥
4. 衡量系统的调度性能主要是 ( ① )  
 ①作业的周转时间      ②作业的输入/输出时间  
 ③作业的等待时间      ④作业的响应时间
5. 批处理系统的主要缺点是 ( ① )  
 ①无交互性      ②输入/输出设备利用率低  
 ③CPU 利用率低      ④失去了多道性
6. 设备处理程序就是 ( ② )  
 ①通道处理程序      ②I/O 进程与设备控制器之间的通信程序  
 ③设备分配程序      ④缓冲管理程序
7. AND 信号量机制是为了 ( ③ )  
 ①信号量的集中使用      ②解决结果的不可再现性问题  
 ③防止系统的不安全性      ④实现进程的相互制约
8. 将硬盘信息存到内存之前的磁盘访问需要三个延迟时间,



请按从最慢到最快的次序，正确选择访问数据的延迟时间。( 4 )

①旋转延迟时间，快表的开销时间，寻道时间

②快表的开销时间，旋转延迟时间，寻道时间

③数据读出或写入磁盘时间，旋转延迟时间，寻道时间

④寻道时间，旋转延迟时间，数据读出或写入磁盘时间

9. 使用访问位的目的是：( 2 )

①实现 LRU 页面置换算法  
置换算法

②实现 NRU 页面  
最近未用算法

③在快表中检查页面是否进入  
最近被写过

④检查页面是否

10. 可能出现抖动的存储管理方法有：( 4 )

①固定式分区别

②动态分区分配

③动态重定位分区分配

④请求分页存储管理

## 六、填空题（每小题 1 分，共 9 分）

1. 某页式存储管理系统中，地址结构的低 9 位表示页内位移量，则页面大小最多为 512 字节。

2. 在多级目录下，对文件的查找可以从 根 目录开始，也可以从 当前工作 目录开始查找。

3. 文件在文件存储空间的组织方式，称为文件的 物理 结构。

4. 一个操作中的所有动作，要么全做，要么全不做，这叫做 原子操作。

5. 要实现程序运行中的结果的可再现性，可采用



bernstein 条件 来保证。

6. 产生死锁的原因是：竞争资源 和 进程推进顺序不当

7. 高级通信方式有：共享存储系统、消息系统 和 管道通信。

8. 作业的控制方式有：脱机作业控制 和 联机作业控制。

9. 通道的类型有：字节多路通道、数组选择通道 和 数组多路通道。

七、判断题（正确的在括号内打√，错误的在括号内打×，每小题 1 分，共 10 分，）

1. ( ☒ ) 在段页式管理中，既有段表又有页表，就不能再使用快表了。

2. ( ☒ ) 实现虚拟存储器的关键技术是提供快速有效地自动地址变换的硬件机构和相应的软件算法。

3. ( ☒ ) 一个操作系统的系统调用越多，系统的功能就越强，用户使用就越复杂。

4. ( ☒ ) 磁盘是共享设备，所以允许多个进程同时在存储空间中进行访问。

5. ( ☒ ) 作业调度程序本身作为一个进程在系统中执行，它是在系统初始化时被创建的。

6. ( ☒ ) 响应比高者优先调度算法解决了长作业死等的问题。

7. ( ☒ ) 必须在分时系统的情况下，进程才能并发执行。

8. ( ☒ ) 检测死锁的算法是按照死锁定理来实现的，必须在死锁发生时调用。检测死锁的算法是在定时检查系统状态时使用。



9. ( ~~X~~ ) 在引入线程的操作系统中, 则把进程作为调度和分派的基本单位, 而把线程作为资源拥有的基本单位。

- ★ 10. ( ~~X~~ ) 虚拟存储器的最大容量是由主存和辅存的容量之和决定的。  
是由地址结构决定的。

### 八、问答题 (每小题 7 分, 共 21 分)

1. 一个单 CPU 的终端, 如果所有用户的 40% 不等待使用终端, 所有用户的平均等待时间是 50 个单位时间, 其它用户的平均等待时间是多少? 设其它用户的等待时间为  $x$ , 则有  
60%  $x$  为其它用户的平均等待时间, 设用户数为  $n$ .

$$50n = n \cdot (1 - 40\%)x \quad x = \frac{250}{3}$$

2. 在一个请求页式存储管理系统中, 采用 LUN 页面置换算法, 假设一进程分配了 4 个页框, 按下列页面运行:

1, 8, 1, 7, 8, 2, 7, 2, 1, 8, 3, 8, 2, 1,

3, 1, 7, 1, 3, 7, 请给出缺页的次数和缺率。

kaoyan.com

[教材 P153]

3. 在一页式存储管理系统中, 访问一个高速的联想存储器中的页面需要 40ns, 并有 80% 成功访问率。而一般的页面的访问需要 100ns, CPU 访问存储器的平均有效时间是多少?

$$\text{成功访问联想时间} = 40 \times 80\% = 3200\%$$

$$\text{普通存储器访问} = 20\% \times 100 = 2000\%$$

$\therefore$  CPU 平均有效访问存储器时间  $R$ :

$$R = \frac{3200 + 2000}{100} = 52$$

$$\text{普通存储器访问} = 20\% \times (100 + 40) = 2800\%$$

$$\therefore R = \frac{3200 + 2800}{100} = 60$$



## 数据结构部分参考答案

### 一、单项选择题 (每题 1 分, 共 8 分)

- |        |        |        |        |
|--------|--------|--------|--------|
| 1. (1) | 2. (1) | 3. (4) | 4. (4) |
| 5. (4) | 6. (3) | 7. (2) | 8. (3) |

### 二、判断题: 正确用√表示, 错误用×表示。(每题 1 分, 共 5 分)

1. (×) 在顺序表中取出第  $i$  个元素所花费的时间与  $i$  成正比。
2. (√) 一个有向图的邻接表和逆邻接表中的结点个数一定相等。
3. (√) 直接选择排序算法的时间复杂度为  $O(n^2)$ , 不受数据初始排列的影响。
4. (×) 由于希尔排序的最后一趟与直接插入排序过程相同, 因此前者一定比后者花费的时间更多。
5. (×) 广义表的长度是指广义表中的原子个数。

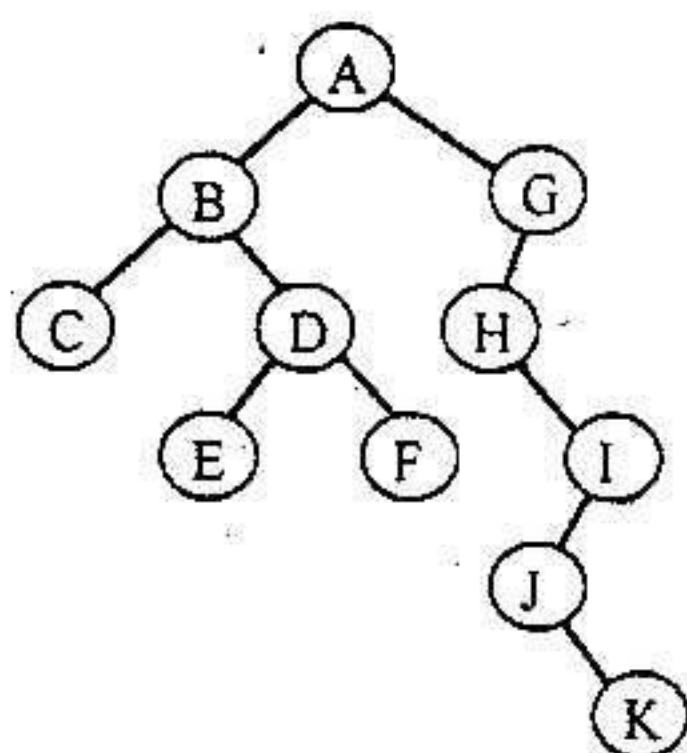
### 三、简答题 (每题 5 分, 共 20 分)

1. 一棵二叉树的先序、中序和后序序列如下, 其中有部分未标出, 试构造出该二叉树。

先序序列为:      CDE      GHI      K

中序序列为: CB           FA      JKIG

后序序列为:      EFDB      JIH      A



2. 将下面算法划线处用具体语句表示, 使其完成所要求的功能。

**PROC** ins\_linklist (la : linklistp; i: integer; b: elemtp);

{la 为带头结点单链表的头指针, 在  $i$  之前插入数据元素  $b$ }

p: = la↑.next; j: = 0;

**WHILE** ( 循环控制条件 ) **DO**

    [ 循环体部分 ];

**IF** ( 判定条件 )

**THEN** **ERROR** (' $i$  非法')

**ELSE** [ new(s); s↑.data: =b;

插入链表部分

    ]

**ENDP**; { ins\_linklist }

**WHILE** (p ( ) nil **AND**  $j < i-1$  ) **DO**

    [ p : = p↑.next; j: =j+1];

**IF** (p=nil **OR**  $j > i-1$  )

**THEN** **ERROR** (' $i$  非法')

**ELSE** [ new(s); s↑.data: =b;

s↑.next: = p↑.next;

p↑.next: = s ]



一棵有  $n$  个顶点的生成树有且仅有  $n-1$  条边。如果它多于  $n-1$  条边，则一定存在环！

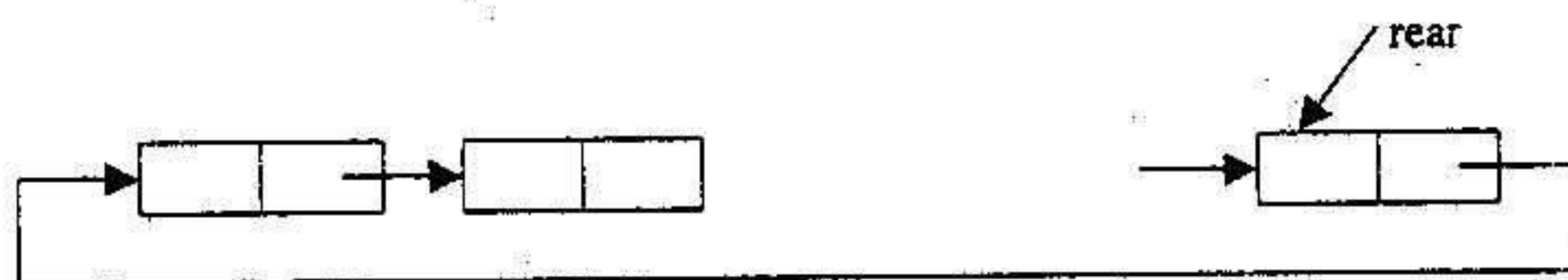
3. 对有向图和无向图，分别描述如何判定图中是否存在回路。

答：对有向图，可以用拓扑排序算法来判定图中是否存在回路。当输出顶点数小于顶点数时，图中存在回路，否则，图中无回路。

对无向图，若边数大于等于顶点数时，则图中存在回路，否则，图中无回路。

4. 试设计一种链队列的存储结构，要求只用一个指针，并且使其插入和删除的时间复杂度均为  $O(1)$ 。试画出存储结构，并指出队首和对尾。

答：采用仅有尾指针  $rear$  的循环单链表表示链队列（如图），队空时  $rear = \text{nil}$ ，队首为第一个结点（ $\text{front} = \text{rear} \uparrow .\text{next}$ ），队尾为最后一个结点（ $\text{rear}$ ）。



#### 四、算法题（共 17 分）

1. 修改下面中序遍历二叉树算法，使其能判定所输出的序列是否有序。（9 分）

```
PROC inorder (bt: bitreptr);
```

```
{bt 为二叉树根结点指针}
```

```
IF bt <> nil THEN
```

```
{ inorder(bt ↑ .lchild);
```

```
visit(bt ↑ .data);
```

```
inorder(bt ↑ .rchild)
```

```
}
```

```
ENDP: { inorder }
```

参考算法：

▲ **算法思想**：用一个变量  $\text{predata}$  作为前趋结点的值，初值为最小值  $-\text{Maxint}$ ，遍历中判断，当某结点的值小于其前趋时，输出序列为无序序列，逻辑变量  $B$  为  $\text{false}$ 。

```
PROC inorder (bt: bitreptr);
```

```
{bt 为二叉树根结点指针，初次调用时  $\text{predata} = -\text{Maxint}$ ， $B$  为  $\text{true}$ }
```

```
IF bt <> nil THEN
```

```
{ inorder(bt ↑ .lchild);
```

```
IF bt ↑ .data < predata THEN B := false;
```

```
predata := bt ↑ .data;
```

```
inorder(bt ↑ .rchild)
```

```
}
```

```
ENDP: { inorder }
```