

题号: 435

## 大连海事大学2005年硕士研究生招生考试试题

考试科目: 数据结构

适用专业: 计算机应用技术、计算机软件与理论

考生须知: 1、所有答案必须写在答题纸上, 写在试题纸上无效;

2、考生不得在答题纸上作与答题内容无关的标记, 否则试卷作废。

共 6 页第 1 页

一、判断下列叙述是否正确。请写出题号并用“√”或“×”回答(共 20 分, 每小题 1 分)

1. 若  $(u, v)$  是连通网络的一条权值最大的边, 则不论采用何种方法构造该网络的最小生成树, 所构造出最小生成树一定不包含  $(u, v)$  这条边。 [ ]

2. 算法是具有有穷性、确定性、可行性、0个或多个输入、1个或多个输出特性的一组规则。操作系统一旦被启动后就永远处在工作或等待状态, 所以, 实现“操作系统”的一组规则不能称为算法。 [ ]

3. 给定  $n$  个不同权值的结点, 则依据这  $n$  个结点构造的 Huffman 树的结构是唯一的。 [ ]

4. 在线索二叉树中, 根据线索可以找到树中任何一个结点在相应遍历序列中的直接前驱或直接后续。 [ ]

5. 在线性表的顺序存储结构中, 每删除一个数据元素都必须移动表中的数据元素。 [ ]

6. 在一个 AOE 网中, 若某一活动的最早开始时间和最迟开始时间相同, 则该活动为关键活动。 [ ]

7. 对有序表而言, 采用折半查找方法查找表中的数据元素, 其查找成功的平均查找长度一定比采用顺序查找方法时的平均查找长度要小。 [ ]

8. 在非空完全二叉树中, 若某结点不存在左孩子, 则该结点一定是叶子结点。 [ ]

9. 设  $L$  是广义表, 则取表头运算  $\text{Head}(L)$  的运算结果一定是单元素, 而取表尾运算  $\text{Tail}(L)$  的运算结果一定是广义表。 [ ]

10. 将一棵树转换成二叉树后, 根结点没有右子树。 [ ]

11. 就平均时间性能而言, 快速排序是最优的。所以, 对于任意的待排序序列, 选择快速排序方法进行排序, 其执行时间将是最少的。 [ ]

12. 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多。 [ ]

13. 存在着这样的非空二叉树, 不论采用怎样的遍历算法其所得到的遍历序列均相同。 [ ]

14. 假设图已经以邻接表存储, 则按深度优先遍历该图所得生成树是唯一的。 [ ]

15. 有向无环图的顶点拓扑排序序列一定是唯一的。 [ ]

16. 健壮算法不会因非法的输入数据而出现莫名其妙的状态。 [ ]

17. 归并排序算法是稳定的排序算法。 [ ]

18. 算法的优劣与所用计算机无关, 也与所用的算法描述语言无关。 [ ]

19. 提高外排序速度的核心工作是减少记录在内外存之间的I/O次数。 [ ]

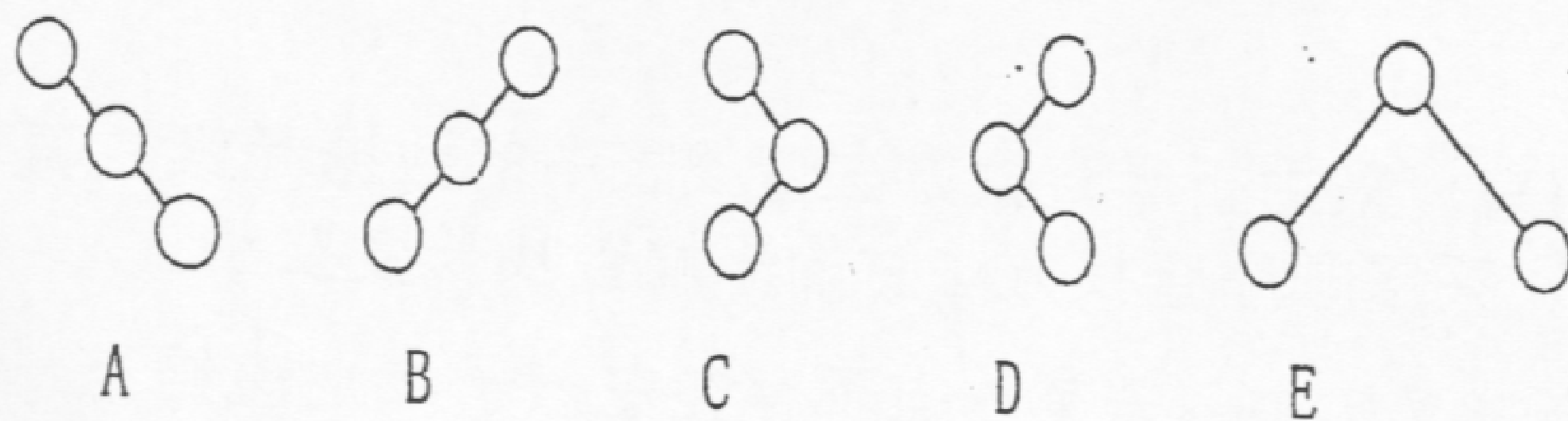
20. 平衡二叉树中所有结点的平衡因子都不超过1。 [ ]

## 二、选择填空。(共 20 分, 每小题 2 分)

1. 设循环队列顺序存放在一维数组  $Sq.data[0..m]$  中, 采用牺牲一个存储单元的方式区分队满与队空的条件, 且假设  $sq.front$  指向队头元素的位置,  $sq.rear$  指向队尾元素的下一个位置, 则判断该队列队满的条件是\_\_\_\_\_。

- A.  $sq.front = (sq.rear + 1) \text{ MOD } m$
- B.  $sq.front = (sq.rear + 1) \text{ MOD } m+1$
- C.  $sq.rear = (sq.front + 1) \text{ MOD } m$
- D.  $sq.rear = (sq.front + 1) \text{ MOD } m+1$
- E.  $sq.rear = (sq.front - 1) \text{ MOD } m+1$

2. 设  $T$  是具有 3 个结点的二叉树, 且  $T$  的后序序列与中序序列相同, 则  $T$  的形态为\_\_\_\_\_。



3. 利用广义表的  $Head(L)$  和  $Tail(L)$  的运算, 将元素  $c$  从广义表  $L = (((a, b), e, (c, d)))$  中分离出来, 其运算表达式为\_\_\_\_\_。

- A.  $Head(Head(Tail(Tail(Head(Head(L))))))$
- B.  $Head(Head(Tail(Tail(Head(L)))))$
- C.  $Head(Tail(Tail(Head(Head(L)))))$

D. Head(Head(Tail(Head(Head(Head(L))))))

E. Head(Head(Tail(Tail(Tail(Tail(L))))))

4. 设四维数组 $B[1..3, 2..8, 0..5, 1..8]$ 以行主序顺序方法存储在一个连续的存储空间内, 每一个数据元素占一个存储单元, 且 $B[1, 2, 3, 4]$ 的存储地址是2000, 则 $B[2, 3, 4, 5]$ 的存储地址是\_\_\_\_\_。

A. 2391

B. 2392

C. 2393

D. 2394

E. 全错

5. 设 $T$ 是一棵平衡二叉树,  $N_h$ 表示深度为 $h$ 的平衡二叉树的最少结点数, 则深度为 $h(h>0)$ 的 $T$ 中最少结点数是\_\_\_\_\_。

A.  $N_{h-1} + N_{h-2} - 2$

B.  $N_{h-1} + N_{h-2} - 1$

C.  $N_{h-1} + N_{h-2} + 2$

D.  $N_{h-1} + N_{h-2} + 1$

E.  $N_{h-1} + N_{h-2}$

6. 若一棵哈夫曼(huffman)树中共有9个顶点, 则其叶子结点数为\_\_\_\_\_个。

A. 3

B. 4

C. 5

D. 6

E. 7

7. 在一棵深度为3的树中, 若有2个度为3的结点和1个为2的结点, 则度为0的结点有\_\_\_\_\_个。

A. 4

B. 5

C. 6

D. 7

E. 3

8. 设结点 $x$ 和 $y$ 是二叉树中的任意两个结点, 在该二叉树的先根遍历序列中 $x$ 在 $y$ 之前, 而在其后根遍历序列中 $x$ 在 $y$ 之后, 则 $x$ 和 $y$ 的关系是\_\_\_\_\_。

A.  $x$ 是 $y$ 左兄弟

B.  $x$ 是 $y$ 右兄弟

C.  $x$ 与 $y$ 没有必然关系

D.  $x$ 是 $y$ 的后裔

E.  $x$ 是 $y$ 的祖先

9. 以下给定的序列中, 不满足堆定义的是\_\_\_\_\_。

A. (97, 87, 93, 79, 82, 62, 84, 42, 22, 12, 68)

B. (97, 93, 87, 84, 82, 79, 68, 62, 42, 22, 12)

C. (97, 87, 42, 79, 82, 62, 68, 93, 84, 12, 22)

D. (12, 22, 42, 62, 68, 79, 82, 84, 87, 93, 97)

E. (97, 93, 87, 79, 82, 62, 84, 42, 22, 12, 68)

10. 若一个具有 $n$ 个结点、 $k$ 条边的非连通无向图是一个森林( $n>k$ ), 则该森林中必有\_\_\_\_\_棵树。

A.  $k$

B.  $n$

C.  $n-k-1$

D.  $n+k$

E.  $n-k$

三、 给定进栈元素顺序 A、B、C、D、E、F、G, 请给出 5 个出栈序列, 其中的 C 必须为第一个出栈元素, F 必须为最后一个出栈元素。(10分)

四、 设主串  $S='abcdabcabababc'$ , 子串  $t='ababab'$ , 求解以下问题。(10分)

(1) 求出模式 $T$ 的Next[]值。

(2) 求出模式 $T$ 的NextVal[]值。



```

printf( r, s );

if ((r <= 0) || (s <= 0)) {
    u = r + s;
} else if (r > s) {
    u = passThis( r - 5, s - 4) + s;
} else {
    u = passThis( r - 4, s - 5) + r;
}
printf( u );
return( u );
}

```

九、 算法 MergeLinkedList 完成两个非空单链表的归并操作。其中头结点指针分别为 La 和 Lb, 表中的元素个数分别为 m 和 n。单链表中存储的数据元素具有可比性且非递增, 归并后的结果为非递增的单链表。阅读该算法并回答下列各问题。 (15 分)

```

Void MergeLinkedList(LinkList &La , LinkList Lb)
//已知单链表La和Lb是非递增的, 将Lb归并到La中且La为非递增的单链表;
pa = La->next; pb = lb->next ; La->next = Null ; p=La ;
While ( pa && pb ) { //①
    if ( pa->data <= pb->data ) {
        S = pa; pa = pa->next; }
    else { S = pb; pb = pb->next; } //②
    ③ ; //在结果单链表中插入结点S。
}; // While
if ( !pa ) {pa = pb;}
While ( pa ) { //④
    S = pa ; pa = pa->next;
    ③ ; //在结果单链表中插入结点S。
} // While
free(Lb);
} // MergeLinkLis

```

问题:

- ①处的合理的注释应该是什么?
- ②处的合理的注释应该是什么?
- ③处遗漏的语句应该是什么?。
- ④处的合理的注释应该是什么?
- 该算法的时间复杂度应该是多少?。

## 十、编程题(15分)

请编写一递归算法 Depth, 计算给定二叉树 Bt 的高度。

编程要求:

BTnode 定义的二叉链结构如下:

```
typedef struct BTnode {
    TelemType data;
    Struct BTnode *rchild, // 右孩子指针
    *lchild; // 左孩子指针
}
```

主程序定义:

```
Int Depth( BTnode *t )
```

递归程序建议:

```
Int subth( BTnode *t )
```

编程提示:

采用后序遍历算法;

成功返回子树高度, 失败返回 -1; 空枝返回 0;

通过比较子树高度, 选择最大者进行后续处理。

## 十一、证明题(15分)

试证明: 对于满k叉树, 其叶子结点数  $n_0$  和非叶子结点数  $n_1$  之间满足如下关系:

$$n_0 = (k - 1) \times n_1 + 1$$