

2000 年南京理工大学数据结构考研试题  
考研加油站收集整理 <http://www.kaoyan.com>

一. 在所给的 A, B, C, D 中选择一个最确切的 (每题 1.5 分, 共 33 分)

1. 下面关于算法说法错误的是

- A) 算法最终必须由计算机程序实现
- B) 为解决某问题的算法同为该问题编写的程序含义是相同的
- C) 算法的可行性是指指令不能有二义性
- D) 以上几个都是错误的

2. 下面说法错误的是

- (1) 算法原地工作的含义是指不需要任何额外的辅助空间
- (2) 在相同的规模  $n$  下, 复杂度  $O(n)$  的算法在时间上总是优于复杂度  $O(2^n)$  的算法
- (3) 所谓时间复杂度是指最坏情况下, 估算算法执行时间的一个上界
- (4) 同一个算法, 实现语言的级别越高, 执行效率就越低

- A) (1) B) (1), (2) C) (1), (4) D) (3)

3. (1) 静态链表既有顺序存储的优点, 又有动态链表的优点。所以, 它存取表中第  $i$  个元素的时间与  $i$  无关。

(2) 静态链表中能容纳的元素个数的最大数在表定义时就确定了, 以后不能增加

(3) 静态链表与动态链表在元素的插入、删除上类似, 不需做元素的移动  
以上错误的是

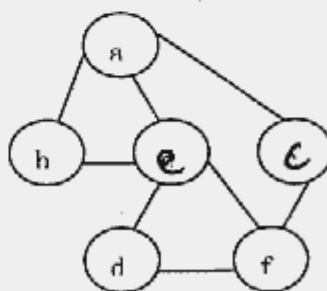
- A) (1)、(2) B) (1) C) (1)、(2)、(3) D) (2)

4. 一棵有  $n$  个结点的二叉树, 按层次从上到下, 同一层从左到右顺序存储在一维数组  $A[1..n]$  中, 则二叉树中第  $i$  个结点 ( $i$  从 1 开始用上述方法编号) 的右孩子在数组  $A$  中的位置是

- A)  $A[2i]$  ( $2i \leq n$ )
- B)  $A[2i+1]$  ( $2i+1 \leq n$ )
- C)  $A[i/2]$
- D) 条件不充分, 无法确定

5. 将有关二叉树的概念推广到三叉树, 则一棵有 244 个结点的完全三叉树的高度是  
A) 4    B) 5    C) 6    D) 7
6. 设栈 S 和队列 Q 的初始状态为空, 元素  $e_1, e_2, e_3, e_4, e_5$  和  $e_6$  依此通过栈 S, 一个元素出栈后即进队列 Q, 若 6 个元素出队的序列是  $e_2, e_4, e_3, e_6, e_5, e_1$ , 则栈 S 的容量至少应该是  
A) 6    B) 4    C) 3    D) 2
7. 对下列四种排序方法, 在排序中关键字比较次数同记录初始排列无关的是  
A) 直接插入    B) 二分法插入    C) 快速排序    D) 归并排序
8. 设树 T 的度为 4, 其中度为 1, 2, 3 和 4 的结点个数分别为 4, 2, 1, 1, 则 T 中的叶子数为  
A) 5    B) 6    C) 7    D) 8
9. 在有向图 G 的拓扑序列中, 若顶点  $V_i$  在顶点  $V_j$  之前, 则下列情形不可能出现的是  
A) G 中有弧  $\langle V_i, V_j \rangle$   
B) G 中有一条从  $V_i$  到  $V_j$  的路径  
C) G 中没有弧  $\langle V_i, V_j \rangle$   
D) G 中有一条从  $V_j$  到  $V_i$  的路径
10. 有数据 {53, 30, 37, 12, 45, 24, 96}, 从空二叉树开始逐个插入数据形成二叉排序树, 若希望高度最小, 则应选择下面哪个序列输入  
A) 45, 24, 53, 12, 37, 96, 30  
B) 37, 24, 12, 30, 53, 45, 96,  
C) 12, 24, 30, 37, 45, 53, 96  
D) 30, 24, 12, 37, 45, 96, 53
11. 有关二叉树下列说法正确的是  
A) 二叉树的度为 2  
B) 一棵二叉树的度可以小于 2  
C) 二叉树中至少有一个结点的度为 2  
D) 二叉树中任何一个结点的度都为 2
12. 设有一个按各元素的值排好序的线性表且长度大于 2, 对给定的值 k, 分别用顺序查找法和二分查找法查找一个与 k 相等的元素, 比较次数分别是 s 和 b, 在查找不成功的情况下, 正确的 s 和 b 的数量关系是  
A) 总有  $s=b$     B) 总有  $s>b$   
C) 总有  $s<b$     D) 与 k 值大小有关

13. 下面关于 B 树和 B<sup>+</sup>树的叙述中, 不正确的结论是
- A) B 树和 B<sup>+</sup>树都能有效的支持顺序查找  
B) B 树和 B<sup>+</sup>树都能有效的支持随机查找  
C) B 树和 B<sup>+</sup>树都是平衡的多分树  
D) B 树和 B<sup>+</sup>树都可用于文件索引结构
14. 某二叉树中序序列为 A, B, C, D, E, F, G, 后序序列为 B, D, C, A, F, G, E, 则前序序列是
- A) E, G, F, A, C, D, B      B) E, A, C, B, D, G, F  
C) E, A, G, C, F, B, D      D) 上面的都不对
15. 上题的二叉树对应的森林包括多少棵树
- A) 1    B) 2    C) 3    D) 概念上是错误的
16. 关于杂凑查找说法不正确的有几个
- (1) 采用链地址法解决冲突时, 查找一个元素的时间是相同的  
(2) 采用链地址法解决冲突时, 若插入规定总是在链首, 则插入任一个元素的时间是相同的  
(3) 用链地址法解决冲突易引起聚集现象  
(4) 再哈希法不易产生聚集
- A) 1    B) 2    C) 3    D) 4
17. 设森林 F 对应的二叉树为 B, 它有 m 个结点, B 的根为 p, p 的右子树结点个数是 n, 森林 F 中第一棵树的结点个数是
- A) m-n    B) m-n-1    C) n+1    D) 条件不足, 无法确定
18. 采用简单选择排序, 比较次数与移动次数分别为
- A)  $O(n), O(\log n)$     B)  $O(\log n), O(n^2)$   
C)  $O(n^2), O(n)$       D)  $O(n \log n), O(n)$
19. 归并排序中, 归并趟数是
- A)  $O(n)$     B)  $O(\log n)$     C)  $O(n \log n)$     D)  $O(n^2)$
20. 设图如右所示, 在下面的 5 个序列中  
符合深度优先遍历的序列有多少
- a e b d f c , a c f d e b, a e d f c b  
a c f d c b , a e f d b c
- A) 5 个    B) 4 个    C) 3 个    D) 2 个



21. (1) 求从指定源点到其余各顶点的迪杰斯特拉 (Dijkstra) 最短路径算法中弧上权不能为负的原因是在实际应用中无意义  
 (2) 利用 Dijkstra 求每一对不同顶点之间的最短路径的算法时间是  $O(n^3)$  (图用邻接矩阵表示)  
 (3) Floyd 求每对不同顶点对的算法中允许弧上的权为负, 但不能有权和为负的回路

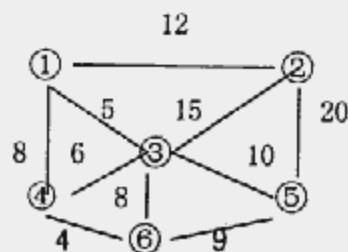
上面不正确的是

- A) (1), (2), (3)    B) (1)    C) (1), (3)    D) (2), (3)
22. (1) 在 AOE-网中, 减小任一关键活动上的权值后, 整个工程的工期也就相应减小  
 (2) AOE-网 工程工期为关键活动上的权之和  
 (3) 在关键路径上的活动都是关键活动, 而关键活动也必在关键路径上
- 上面有错的说法个数是
- A) (1)    B) (2)    C) (3)    D) (1), (2)

## 二、填空 (共 33 分, 每空 1.5 分, 答题时注明填空补缺处的序号, 不需注题号)

1. 计算机执行下面的语句时, 语句 s 的执行次数为 (1)。
- ```
for(i=1; i<n-1; i++)
  for(j=n; j>=i; j--)
    s;
```
2. 对于双向链表, 在两个结点之间插入一个新结点需修改的指针共 (2) 个, 单链表为 (3) 个。
3. 求从某源点到其余各顶点的 Dijkstra 算法在图的顶点数为 10, 用邻接矩阵表示图时计算时间约为 10ms, 则在图的顶点数为 40 时, 计算时间约为 (4) ms。
4. 向一棵 m 阶 B-树插入操作时, 当结点的关键字数为 (5) 则要分裂该结点; 删除时, 结点中关键字数为 (6) 时, 可能需要同左或右兄弟合并。
5. 按 13, 24, 37, 90, 53 的次序形成二叉平衡树, 则该二叉平衡树的高度是 (7), 其根为 (8), 左子树中的数据是 (9), 右子树中的数据是 (10)。

6. 对于右面的图, 用 Prim 算法从顶点①开始求最小生成树, 按次序产生的边为 (11) 共 (12) 条; 用 Kruskal 算法产生的边的次序是 (13)。  
 (注意: 边用 (i, j) 的形式表示)



7. 已知有序表为 (12, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134) 当用二分法查找 90 时,

需 (14) 次查找成功, 47 时 (15) 成功, 查 100 时, 需 (16) 次才能确定不成功。

8. 设一棵后序线索树的高是 50, 结点  $x$  是树中的一个结点, 其双亲是结点  $y$ ,  $y$  的右子树高度是 31,  $x$  是  $y$  的左孩子。则确定  $x$  的后继最多需经过 (17) 中间结点 (不含后继及  $x$  本身)

9. 设  $F$  是由  $T_1, T_2, T_3$  三棵树组成的森林, 与  $F$  对应的二叉树为  $B$ , 已知  $T_1, T_2, T_3$  的结点数分别为  $n_1, n_2$  和  $n_3$ , 则二叉树  $B$  的左子树中有 (18) 个结点, 右子树中有 (19) 个结点。

10. 设有三对角矩阵如下图

$$A = \begin{pmatrix} a_{11} & a_{12} & & & & \\ & a_{21} & a_{22} & a_{23} & & \\ & & a_{32} & a_{33} & a_{34} & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1,n} & \\ & & & & & a_{n,n-1} & a_{nn} \end{pmatrix}$$

将带状区域中的元素  $a_{i,j}$  ( $|i-j| \leq 1$ ) 放在一维数组  $B$  中, 则  $B$  的大小为 (20), 元素  $a_{i,j}$  在  $B$  中的位置是 (21) (下标从 0 开始计)

11. 三维数组  $a[4][5][6]$  (下标从 0 开始计,  $a$  有  $4 \times 5 \times 6$  个元素), 每个元素的长度是 2, 则  $a[2][3][4]$  的地址是 (22)。(设  $a[0][0][0]$  的地址是 1000, 数据以行为主方式存储)

三、将下面的类 C 写的程序补充完整, 凡未说明的变量均是工作变量 (共 34 分, 每空 1 分, 第 5 小题为 3+3 分, 答题时注明填空缺处的序号, 不需注题号)

1. 下面是中序线索树的遍历算法, 树有头结点且由指针  $thr$  指向。树的结点有五个域, 分别为: 数据域  $data$ , 左、右孩子域  $lchild, rchild$  和左、右标志域  $ltag, rtag$ , 规定, 标志域为 1 是线索, 0 是指向孩子的指针。

inordethread( $thr$ )

```
{
    p = thr->lchild;
    while( (1) ) {
        while( (2) )
            p = (3);
        printf(p->data);
        while( (4) ) {
```

```

    p= (5) :
    printf(p->data); }
    p= (6) :
    }
}

```

2. 用链表表示的数据的简单选择排序，结点的域为数据域 data，指针域 next；链表首指针为 head，链无头结点。

```

selectsort(head)
{
    p=head;
    while(p (7) ) {
        q=p;
        r= (8) ;
        while( (9) ) {
            if( (10) )
                q=r;
            r= (11) ;
        }
        tmp=q->data;
        q->data=p->data;
        p->data=tmp;
        p= (12) ;
    }
}

```

3. 一元稀疏多项式以循环单链表按降幂排列，结点有三个域，系数域 coef，指数域 exp 和指针域 next；现对链表求一阶导数，链表的头指针为 ha，头结点的 exp 域为-1。

```

derivative(ha)
{
    q=ha;
    pa=ha->next;
    while( (13) )
    {
        if( (14) ) {
            (15) ;
            free(pa);
            pa= (16) ;
        }
        else {
            pa->coef (17) ;
        }
    }
}

```

```

    pa->exp__(18);
    q=__(19)__;
    pa=__(20)__;
}
}

```

4.  $n$  个顶点的有向图用邻接矩阵 `array` 表示, 下面是其拓扑排序算法, 试补充完整

注: (1) 图的顶点号从 0 开始计

(2) `indegree` 是有  $n$  个分量的一维数组, 放顶点的入度。

(3) 函数 `crein` 用于算顶点入度

(4) 有三个函数 `push(data)`, `pop()`, `check()` 其含义为数据 `data` 进栈, 退栈和测试栈是否空 (不空返回 1, 否则 0)

(5) 算法中每行后的双斜杠加数字为语句行号, 在下题中使用, 与本题无关。

```

crein(array, indegree, n)                                //1
{   //2
    for(i=0; i<n; i++)                                   //3
        indegree[i]=__(21)__;                             //4
    for(i=0; i<n; i++)                                   //5
        for(j=0; j<n; j++)                               //6
            indegree[i]+=array[(22)][__(23)];             //7
   //8

```

```

topsort(array, indegree, n)                              //1
{   //2
    count=__(24)__;                                       //2
    for(i=0; i<n; i++)                                   //3
        if(__(25)__)                                     //4
            push(i);                                     //5
    while(check())                                       //6
    {
        vex=pop();                                       //7
        printf(vex);                                    //8
        count++;   //9
        for(i=0; i<n; i++)                               //10
        {   //11
            k=array__(26)__;                             //12

```



```

        if( (27) ) //13
        { //14
            indegree[i]--; //15
            if( (28) ) //16
                push(i); //17
        } //18
    } //19
} //20
if( count<n ) //21
    printf("图有回路") //22
} //23

```

5. 在前题的基础上进行求关键路径的第一步，即算出各顶点的最早发生时间（记在  $ve[k]$  中， $0 \leq k \leq n-1$ ），为此，邻接矩阵定义为

$$G[i, j] = \begin{cases} a_{i,j} \\ 0 \end{cases}$$

其中  $a_{i,j}$  是弧  $\langle i, j \rangle$  上的权，若不存在弧  $\langle i, j \rangle$ ，则为 0（ $a_{i,i}$  规定为 0）

在上述条件下，除了将数组  $ve$  作为  $topsort$  的参数， $topsort(array, indegree, ve, n)$  外

则（1） $crein$  要否改，如何改（写出修改部分的语句）（3分）

（2） $topsort$  如何修改，写出修改部分语句（3分）

注意：若需删去某语句注明语句号；插入语句注明在哪一行后插入；  
若修给某行，注明行号替换为. . . .