

机密★启用前

重 庆 邮 电 大 学

2018 年攻读硕士学位研究生入学考试试题

科目名称： 数据结构

科目代码： 802

考生注意事项

- 1、答题前，考生必须在答题纸指定位置上填写考生姓名、报考单位和考生编号。
- 2、所有答案必须写在答题纸上，写在其他地方无效。
- 3、填（书）写必须使用 0.5mm 黑色签字笔。
- 4、考试结束，将答题纸和试题一并装入试卷袋中交回。
- 5、本试题满分 150 分，考试时间 3 小时。

一、选择题（本大题共 15 小题，每小题 2 分，共 30 分）

1. 下面程序段的时间复杂度是（ ）。

```
i = 1;
while (i <= n)
    i = i * 3;
```

A. $O(n)$ B. $O(n \log(n))$ C. $O(\log(n))$ D. $O(\log_3 n)$

2. 在 n 个元素的顺序表中插入或删除一个元素，需要平均移动表中（ ）个元素。

A. (n) B. $(n/2)$ C. (n^2) D. (1)

3. 设循环队列中数组的下标范围是 $0, \dots, m-1$ ，其头指针 $front$ 指向队首元素， $rear$ 指向队尾元素，则队列的长度为（ ）。

A. $(rear - front + 1) \% (m + 1)$ B. $(rear - front + m + 1) \% m$
C. $rear - front$ D. $rear - front + 1$

4. 设计一个十进制转换为八进制的算法，采用（ ）数据结构最佳。

A. 栈 B. 队列
C. 顺序结构线性表 D. 链式结构线性表

5. 若某个栈的输入序列为 $1, 2, 3, \dots, n$ ，输出序列的第一个元素为 n ，则第 i 个输出元素为（ ）。

A. i B. $n-i$ C. $n-i+1$ D. 哪个元素无所谓

6. 六个元素按 $6, 5, 4, 3, 2, 1$ 的顺序进栈，下列哪个出栈序列是错误的（ ）。

A. 543612 B. 453126
C. 346521 D. 234156

7. 某二叉树的先序序列和后序序列正好相反，则该二叉树一定是（ ）二叉树。

A. 空或只有一个结点 B. 高度等于其结点数
C. 任一结点无左孩子 D. 任一结点无右孩子

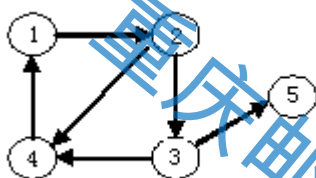
8. 高度为 k 的完全二叉树至少有（ ）个结点（空树高度为 0）。

A. 2^{k-1} B. 2^k
C. $2^k - 1$ D. k

9. 设高度为 h 的二叉树上只有度为 0 和度为 2 的结点，则此二叉树中至多有（ ）个结点。

A. $2^h - 1$ B. 2^{h-1} C. $2^h + 1$ D. $2^{h+1} - 1$

10. 数组 A 中，每个元素的长度为 3 个字节，行下标 i 从 1 到 8，列下标 j 从 1 到 10，从首地址 SA 开始连续存放在存储器内，该数组按行优先存放时，元素 A[8][5] 的起始地址为 ()。
- A. SA+141 B. SA+222
C. SA+144 D. SA+225
11. 任何一个无向连通图的最小生成树 ()。
- A. 有一棵或多棵 B. 一定只有一棵
C. 一定有多棵 D. 可能不存在
12. 对于一个具有 n 个顶点和 e 条边的无向图，若采用邻接表表示，则表头向量的大小为 n；所有邻接表中的结点总数是 ()。
- A. e/2 B. e C. 2e D. n+e
13. 设结点 x 和结点 y 是二叉树 T 中的任意两个结点，若在先序序列中 x 在 y 之前，而在后序序列中 x 在 y 之后，则 x 和 y 的关系是 ()
- A. x 是 y 的左兄弟 B. x 是 y 的右兄弟
C. x 是 y 的祖先 D. x 是 y 的后代
14. 关于下面的图形，哪个说法正确 ()。



- A. 路径 $\langle 1, 2 \rangle, \langle 2, 4 \rangle, \langle 4, 1 \rangle$ 是一条回路；
B. 顶点 2 的入度为 2；
C. 顶点 4 的出度为 2；
D. 以上皆非。
15. 下列序列中，() 是执行第一趟快速排序后得到的序列（排序的关键字类型是字符串）。
- A. [da, ax, eb, de, bb] ff [ha, gc] B. [cd, eb, ax, da] ff [ha, gc, bb]
C. [gc, ax, eb, cd, bb] ff [da, ha] D. [ax, bb, cd, da] ff [eb, gc, ha]

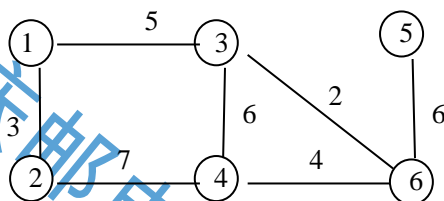
二、填空题（本大题共 10 小题，每小题 3 分，共 30 分）

- 采用顺序查找方法查找长度为 n 的线性表时，在等概率情况下查找成功的平均查找长度为_____。
- 已知数据表 A 中每个元素距其最终位置不远，则采用_____排序算法最节省时间。
- 图 G 是一个非连通图，共有 28 条边，则该图至少有_____个顶点。
- 设一循环队列 Q 中，rear 指针指向队尾元素的下一个位置，front 指针指向队首元素，则判断队列中元素为空的条件是_____。

5. 在大根堆中, 关键字最小的元素可能存放在堆的任一_____结点上。
6. 某后缀表达式为 $abcd-*+ef/-$, 令 $a=2, b=3, c=4, d=5, e=6, f=2$, 则该表达式的值等于_____。
7. 有 n 个顶点的连通图用邻接矩阵表示时, 该矩阵至少有_____个非零元素。
8. 高度 (空树高度为 0) 为 5 的 AVL 树, 其结点数最少是_____。
9. 广义表 $((a), ((b), c), (((d))))$ 的长度是_____, 深度是_____。
10. 在有 n 个结点的二叉链表中, 空链域的个数为_____。

三、问答题 (本大题共 6 小题, 每小题 10 分, 共 60 分)

1. 已知二叉树的先序序列和中序序列分别为 ABDGCEFH 和 DGBAECFH:
 - (1) 画出该二叉树;
 - (2) 写出此二叉树的后序序列;
 - (3) 画出与此二叉树对应的森林。
2. 图 G 各顶点的连接关系及相应权值如下图所示:
 - (1) 画出图的邻接矩阵存储图示;
 - (2) 从顶点 1 开始对图进行广度优先遍历, 写出遍历结果;
 - (3) 使用 Kruskal 算法求该图的最小生成树, 给出形成过程。



3. 设散列表的长度为 8, 散列函数 $H(k)=k \bmod 7$, 初始记录关键字序列为 (25, 31, 8, 27, 13, 68), 要求:
 - (1) 分别给出用线性探测法和链地址法作为解决冲突方法的过程;
 - (2) 计算 (1) 中两种解决冲突方法的平均查找长度。
4. 已知一个图的顶点集 V 和边集 E 分别为:

$$V=\{1,2,3,4,5,6,7\};$$

$$E=\{ \langle 2,1 \rangle, \langle 3,2 \rangle, \langle 3,6 \rangle, \langle 4,3 \rangle, \langle 4,5 \rangle, \langle 4,6 \rangle, \langle 5,1 \rangle, \langle 5,7 \rangle, \langle 6,1 \rangle, \langle 6,2 \rangle, \langle 6,5 \rangle \};$$
 若存储它采用邻接表, 并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的,

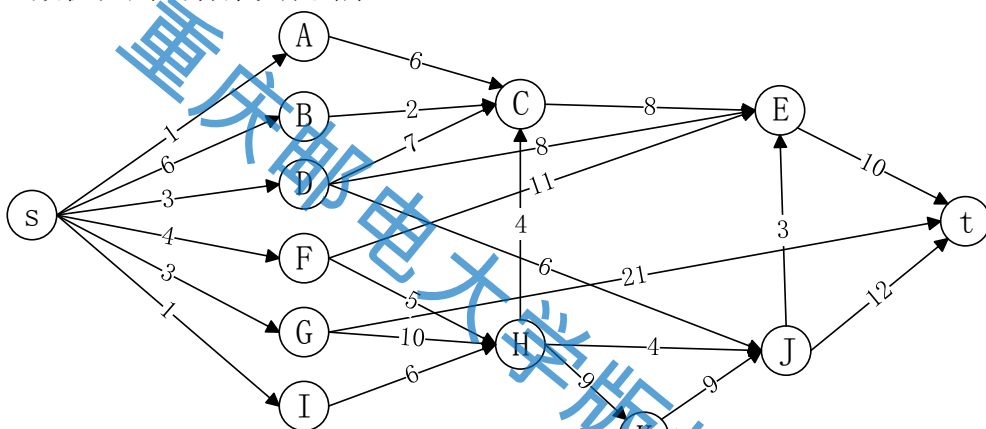
- (1) 画出该图的邻接表存储图示；
- (2) 按拓扑排序算法进行排序，试给出得到的拓扑排序的序列。

5. 假设一个线性链表的类名为 `linkedList`，链表结点的类名为 `ListNode`，它包含两个数据成员 `data` 和 `link`。`data` 存储该结点的数据，`link` 是链接指针。下面给定一段递归打印一个链表中所有结点中数据的算法：

```
void PrintList (ListNode *L) {
    if (L != NULL) {
        printf("%d ", L->data);
        PrintList ( L->link );
    }
}
```

试问此程序在什么情况下不实用？给出具体修改后的可实用的程序？

6. 对于如下图所示的 AOE 网络，计算各活动弧的 $e(a_i)$ （活动 a_i 的最早开始时间）和 $l(a_j)$ （活动 a_j 的最迟开始时间）函数值、各事件（顶点）的 $ve(v_i)$ （事件 v_i 的最早发生时间）和 $vl(v_j)$ （事件 v_j 的最迟发生时间）函数值；列出各条关键路径。



四、问答题（本大题共 2 小题，每小题 15 分，共 30 分）

1. 现有关键字序列{45, 24, 37, 53, 12, 93, 47, 60}，按以下要求完成：
 - (1) 根据给定的关键字序列构造一棵二叉查找（排序）树，以二叉链表形式存储，进行中序遍历可以得到从小到大排列的有序序列，请写出构造过程（不要求算法）。

- (2) 在(1)的基础上, 请编写一个函数(`int LeafCount(Binary_tree BT)`), 求此二叉树的叶子结点个数。

有关的数据结构已描述如下:

```
typedef struct { //二叉树结点
    int data;
    Binary_node *left;
    Binary_node *right;
} Binary_node, *Binary_tree;

int LeftCount ( Binary_tree bt ); //计算树bt的叶结点的个数
```

2. 下面给出一个排序算法, 其中 n 是数据类型为 `Type` 的数组 `A[]` 中元素总数。

```
void unknown (Type a[ ], int n) {
    int d = 1, j;
    while ( d < n / 3 ) d = 3*d+1;
    while ( d > 0 ) {
        for ( int i = d; i < n; i++ ) {
            Type temp = a[i];
            j = i;
            while ( j >= d && a[j-d] > temp ) { a[j] = a[j-d]; j -= d; }
            a[j] = temp;
        }
        d /= 3;
    }
}
```

- (1) 阅读此算法, 说明它的功能;
- (2) 对于下面给出的整数数组, 追踪第一趟 `while (d > 0)` 内的每次 `for` 循环结束时数组中数据的变化。(为清楚起见, 本次循环未涉及的不移动的数据可以不写出, 每行仅写出一个 `for` 循环的变化);

| 步 | a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] | 移动次数 |
|-----|------|------|------|------|------|------|------|------|------|------|------|
| | 77 | 44 | 99 | 66 | 33 | 55 | 88 | 22 | 44 | 11 | |
| 1 | | | | | | | | | | | |
| 2 | | | | | | | | | | | |
| ... | | | | | | | | | | | |

- (3) 以上各次循环的数据移动次数分别是多少。