



暨南大学  
JINAN UNIVERSITY

2019 年全国硕士研究生统一入学考试自命题试题 (A 卷)

\*\*\*\*\*

招生专业与代码: 计算机科学与技术、软件工程、网络空间安全、工程硕士

研究方向: 计算机系统结构 081201, 计算机软件与理论 081202, 计算机应用技术 081203,

软件工程 083500, 计算机技术(专业学位) 085211, 网络空间安全 083900

考试科目名称及代码: 数据结构 830

考生注意: 所有答案必须写在答题纸(卷)上, 写在本试题上一律不给分。

一、单项选择题(每题 2 分, 共 30 分)

1. 在任意一棵二叉树的先序序列和后序序列中, 各叶子之间的相对次序关系 ( )。  
A. 不一定相同      B. 互为逆序      C. 都不相同      D. 都相同
2. 深度为 4 的二叉树至多有结点数为 ( )。  
A. 18      B. 14      C. 15      D. 16
3. 在一个具有  $n$  个顶点的有向图中, 若所有顶点的入度数之和为  $m$ , 则所有顶点的度数之和为 ( )。  
A.  $m$       B.  $m-1$       C.  $m+1$       D.  $2m$
4. 快速排序在( )情况下最不利于发挥其长处。  
A. 被排序的数据量太大.      B. 被排序数据中含有多个相同的关键字  
C. 被排序的数据完全无序      D. 被排序的数据已基本有序
5. 一组记录的关键字为 (45,80,55,40,42,85), 则利用堆排序的方法建立的初始堆为 ( )。  
A. (80,45,55,40,42,85)      B. (85,80,55,40,42,45)  
C. (85,80,55,45,42,40)      D. (85,55,80,42,45,40)
6. 对有 18 个元素的有序表(下标为 1~18)作折半查找,则查找 A[3]的比较序列的下标为 ( )。  
A. 1,2,3      B. 9,5,2,3      C. 9,5,3      D. 9,4,2,3
7. 具有  $n$  个顶点的完全有向图的边数为( )。  
A.  $n(n-1)/2$       B.  $n(n-1)$       C.  $n^2$       D.  $n^2-1$
8. 利用逐点插入法建立序列 (50,72,43,85,75,20,35,45,65,30) 对应的二叉排序树以后, 查找元素 35 要进行 ( )。  
A. 4 次      B. 5 次      C. 3 次      D. 2 次
9. 判定一个有向图是否存在回路除了可以利用拓扑排序方法外, 还可以用 ( )。  
A. 求最短路径的 Floyd 方法      B. 求最短路径的 Dijkstra 方法  
C. 广度优先遍历算法      D. 深度优先遍历算法
10. 对于一个具有  $n$  个顶点的无向连通图, 它包含的连通分量的个数为 ( )。  
A. 0      B. 1      C.  $n$       D.  $n+1$
11. 在一个单链表中,若  $p$  所指的结点不是最后一个结点,在  $p$  之后插入  $s$  所指的结点, 则执行 ( )。  
A.  $s->next=p; p->next=s$       B.  $p->next=s; s->next=p$   
C.  $p=s; s->next=p->next$       D.  $s->next=p->next; p->next=s$

12. 设 F 是由 T1、T2 和 T3 三棵树组成的森林，与 F 对应的二叉树为 B，T1、T2 和 T3 的结点数分别为  $N_1$ 、 $N_2$  和  $N_3$ ，则二叉树 B 的根结点的左子树的结点数为（ ）。
- A.  $N_1-1$       B.  $N_2-1$       C.  $N_2+N_3$       D.  $N_1+N_3$
13. 设输入元素 1,2,3,P,A，输入次序为：123PA，元素经过栈后到达输出序列。当所有元素均达到输出序列，下面（ ）序列可以作为高级语言的变量名。
- A. 123PA      B. PA321      C. 12AP3      D. PA123
14. 在一个链队列 Q 中，删除一个结点需要执行的指令是（ ）。
- A.  $Q.rear=Q.front->next;$       B.  $Q.rear->next=Q.rear->next->next;$   
 C.  $Q.front->next=Q.front->next->next;$       D.  $Q.front=Q.rear->next;$
15. 如果 T2 是由树 T 转换而来的二叉树,那 T 中结点的后序就是 T2 中结点的（ ）。
- A. 先序      B. 中序      C. 后序      D. 层次序

## 二. 填空题(每空 2 分, 共 20 分)

- 设根结点在第一层，那么具有  $n$  个结点的完全二叉树，其高度为\_\_\_\_\_。
- 对于一个循环队列  $Q[0..m-1]$ ，队头、队尾指针分别为  $f$ 、 $r$ ，其判空的条件是\_\_\_\_\_，判满的条件是\_\_\_\_\_。
- 在堆排序，希尔排序，快速排序，归并排序算法中，占用辅助空间最多的是\_\_\_\_\_。
- 已知二维数组  $A[m][n]$  采用行序为主序存储，每个元素占  $k$  个存储单元，并且第一个元素的存储地址是  $Loc(A[0][0])$ ，则  $A[i][j]$  的地址是\_\_\_\_\_。
- 若某记录序列的关键字序列是 (235, 346, 021, 558, 256)，用链式基数排序方法排序，第一次收集的结果是\_\_\_\_\_。
- 设 Hash 表为  $m=11$ ，散列函数  $H(k)=k\%11$ ，表中已有 4 个结点，地址分别为： $addr(15)=4, addr(38)=5, addr(61)=6, addr(84)=7$ ，其余地址为空。如果用二次探测再散列处理冲突，关键字为 49 的结点的地址是\_\_\_\_\_。
- 在一个 3 阶的 B-树上，每个结点包含的子树相同，最多为\_\_\_\_\_个结点，最少为\_\_\_\_\_个结点。
- 一个连通图的\_\_\_\_\_ 是一个极小连通子图。

## 三. 判断题 (每题 1 分, 共 10 分, 正确的选 t, 错误的选 f)

- 对于  $n$  个记录的集合进行冒泡排序，在最坏情况下的时间复杂度是  $O(n^2)$ 。( )
- 包含两个结点的所有二叉树都是相同的。( )
- 一个图按广度优先遍历的结果是唯一的。( )
- 用 Prime 算法和 Kruskal 算法求得的最小生成树一定相同。( )
- 线性表中的每一个元素都有一个前驱和后继元素。( )
- 在  $n$  个顶点的无向图中，若边数  $> n-1$ ，则该图必是连通图。( )
- 完全二叉树的某结点若无左孩子，则必是叶子结点。( )
- 在 B-树,有  $n$  棵子树的结点中有  $n$  个关键字。( )
- 在一个有向图的邻接表或逆邻接表中，如果某个顶点的链表为空，则该顶点的度一定为零。( )
- 用邻接矩阵法存储一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小与图中结点的个数有关，而与图的边数无关。( )

#### 四. 简答题 (40 分)

1. 设  $G$  为有  $n$  个顶点的无向连通图, 证明  $G$  至少有  $n-1$  条边。 (7 分)
2. 什么是线索二叉树? 一棵二叉树的中序遍历序列为  $djbaechif$ , 前序遍历序列为  $abdjcefh$ , 请画出该二叉树的后序线索二叉树。 (7 分)
3. 已知某通讯电文仅有  $A$ 、 $B$ 、 $C$ 、 $D$ 、 $E$ 、 $F$  六个字符构成, 其出现频率分别为  $23, 5, 14, 8, 25, 7$ , 请给出他们的 Huffman 编码以及求解过程。 (7 分)
4. 给定一棵二叉链表存储的二叉树, 试用文字描述判定一棵二叉树是否是完全二叉树的算法基本思想。 (7 分)
5. 已知一棵完全二叉树共有 67 个结点, 试求: (7 分)
  - (1) 树的深度;
  - (2) 度为 1 的结点数;
  - (3) 叶子结点数;
6. 对给定的一组关键字序列  $(29, 18, 25, 47, 58, 12, 51, 10)$ , 写出用归并排序方法进行排序的变化过程。 (5 分)

#### 五. 算法填空 (共 2 小题, 每空 2 分, 共 20 分)

1. 若二叉排序树  $T$  中存在其关键字等于  $key$  的数据元素时, 则下面算法删除该数据元素结点, 并返回  $TRUE$ ; 否则返回  $FALSE$ 。请在\_\_\_\_\_处填上适当内容, 使其成为一个完整算法。

```
typedef struct BiTNode {  
    TElemType    data;  
    struct BiTNode *lchild, *rchild;  
} BiTNode, *BiTree;
```

```
Status DeleteBST(BiTree &T, KeyType key) {  
    if (!T) return FALSE;  
    else {  
        if (EQ(key, T->data.key))    return Delete(T);  
        else if (LT(key, T->data.key)) return DeleteBST(T->lchild, key);  
        else return DeleteBST(T->rchild, key);  
    }  
}
```

```

Status Delete(BiTree &p) {
    BiTree q, s;
    if (!p->rchild) {
        q = p; (1) free(q);
    }
    else if (!p->lchild) {
        q = p; (2) free(q);
    }
    else {
        q = p; (3) ;
        while ( (4) ) { q = s; s = s->rchild; }
        p->data = s->data;
        if ( q != p ) q->rchild = s->lchild;
        else (5)
        free(s);
    }
    return TRUE;
}

```

2. 下面是快速排序算法。请在\_\_\_\_\_处填上适当内容，使其成为一个完整算法。

```

#define Maxsize 100
typedef int keytype;
typedef struct
{
    keytype key;
    Infotype otherinfo;
} RedType;
typedef struct
{
    RedType r[Maxsize+1];
    int length;
} SqList;

void Qsort(SqList &L, int low, int high)
{
    if (low < high)
    {
        pivotloc = Partition(L, low, high);
        (6);
        Qsort(L, pivotloc+1, high);
    }
}

```

```

int Partition(Sqlist &L,int low,int high) {
    L.r[0]=L.r[low];
    pivotkey=L.r[low].key;
    while ( low<high ) {
        while (low<high&&L.r[high].key>=pivotkey) __(7)__;
        L.r[low]=L.r[high];
        while (low<high&&L.r[low].key<=pivotkey) __(8)__;
        L.r[high]=L.r[low];
    }
    L.r[low]=__(9)__;
    return __(10)__;
}

```

## 六. 编写算法 (30 分)

1. 试编写算法，从大到小输出二叉排序树中所有的值不小于  $x$  的关键字。(10 分)
2. 设有一个由正数组成的无序单链表，试编写算法实现下列功能：（10 分）
  - (1) 找出最小值结点，并输出该数值；
  - (2) 若该最小值是奇数，则将其与直接后继结点的数值交换；如该最小值是偶数，则将其直接后继结点删除。
3. 编写一个算法根据用户输入的偶对（以输入 0 表示结束）建立其有向图的邻接表（设有  $n$  个顶点）。（10 分）