

---

# 安徽大学2005-2006学年第1学期

## 《 数据结构 》期末考试试卷 (A卷)

**一、单项选择 (在备选答案中选出一个正确答案，并将其号码填在题后的括号内。每题2分，共20分)**

01. 以下序列不是堆的是( )。  
A. (100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66)    B. (100, 98, 85, 82, 80, 77, 66, 60, 40, 20, 10)  
C. (10, 20, 40, 60, 66, 77, 80, 82, 85, 98, 100)    D. (100, 85, 40, 77, 80, 60, 66, 98, 82, 10, 20)
02. 当采用分块查找时，数据的组织方式为 ( )  
A. 数据分成若干块，每块内数据有序  
B. 数据分成若干块，每块内数据不必有序，但块间必须有序，每块内最大（或最小）的数据组成索引块  
C. 数据分成若干块，每块内数据有序，每块内最大（或最小）的数据组成索引块  
D. 数据分成若干块，每块（除最后一块外）中数据个数需相同
03. 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. 以上答案都不对
04. 有n个叶子的哈夫曼树的结点总数为 ( )。  
A.  $2n-1$     B.  $2n$     C.  $2n+1$     D. 不确定
05. 有一个具有n个顶点的连通图生成的最小生成树中，具有( )条边  
A. n    B. n-1    C. n+1    D.  $2n-1$
06. 下面的二叉树中，( ) 不是平衡二叉树。

A

B

C

D

07. 知U= ‘xyxyxyxxxxy’； t= ‘xxy’；依次执行下列运算后，ASSIGN (S, U)；ASSIGN (V, SUBSTR (S, INDEX (s, t), LEN (t) +1))；ASSIGN (m, ‘ww’），则REPLACE (S, V, m) 结果为( )。  
A. ’xyxywwwxy’    B. ’xyxyxywwy’  
C. ’xyxyxwwxy’    D. ’xyxywwxyx’
08. 设有数组A[i, j]，数组的每个元素长度为5字节，i的值为0 到5，j的值为0 到6，数组从内存首地址BA开始顺序存放，当用以行为主序存放时，元素A[5, 5]的存储首地址为( )。  
A. BA+180    B. BA+175    C. BA+200    D. BA+205
09. 依次读入数据元素序列{a, b, c, d, e, f}进栈，每进一个元素机器可要求下一个元素进栈和出栈。如此进行，则栈空时弹出的元素构成的序列不可能出现( )  
A. {c, d, b, e, f, a}  
B. {d, c, e, b, f, a}  
C. {b, d, c, e, a, f}  
D. {b, e, d, a, c, f}
10. 从具有n 个结点的二叉排序树中查找一个元素时，在最坏情况下的时间复杂度为( )  
A. O(n)    B. O(1)    C. O(logn)    D. O( $n^2$ )

**二、填空题 (每空2分，共20分)**

1. 与链式存储结构不同，顺序存储结构是通过\_\_\_\_\_表示元素之间的逻辑关系的。
2. 若二叉树采用二叉链表存储结构，要交换其所有分支结点左、右子树的位置，利用\_\_\_\_\_遍历方法最合适。
3. 在完全二叉树中，编号为i和j的两个结点处于同一层的条件是\_\_\_\_\_。
4. 对于17个元素的有序表A[1]-A[17]作二分查找，在查找其等于A[8]的元素时需比较\_\_\_\_次。
5. 树的三种存储结构是 双亲表示法、孩子表示法和\_\_\_\_\_。
6. 若不考虑基数排序，则在内部排序过程中，主要进行的两种基本操作是关键字的比较和记录的\_\_\_\_\_。

7. 一个“好”的算法要考虑以下标准：正确性、可读性、\_\_\_\_\_和效率与低存储量需求。

8. 已知一个无向图的邻接表如下图所示：

则从顶点1出发进行深度优先搜索遍历

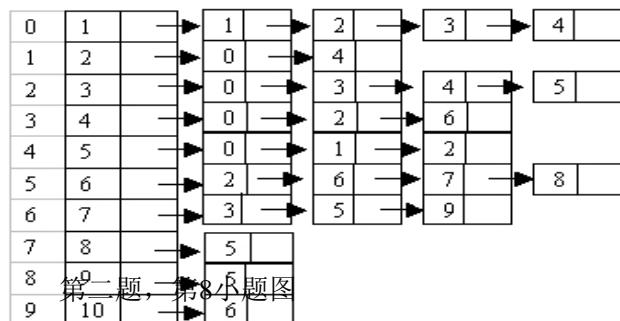
得到的顶点序列为\_\_\_\_\_和广度

优先搜索得到的顶点序列为\_\_\_\_\_

\_\_\_\_\_。

9. 在含有n个结点的二叉链表

中，其空链域个数是\_\_\_\_\_。



### 三、分析计算题 (1、3、5每题8分, 2题6分)

1. 对于如下的连通图，请给出从顶点0出发，利用普里姆(Prim)算法求出它的最小生成树的过程中得到的顶点集和边集及最小生成树的权，并画出所得到的最小生成树。

12

6 8      15 13

16

4

12 9 20 10

5

(第三大题第1题图)

2. 设 $F=\{T_1, T_2, T_3\}$ 是森林(如下图所示)，试将它转换为二叉树，画出所对应的二叉树。

T1            T2            T3

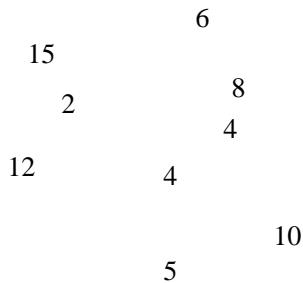
森林 (第三大题第2题图)

3. 记录的关键字集合 $K=\{23, 9, 39, 5, 68, 12, 62, 48, 33\}$ ，请给出采用快速排序法进行排序时每趟划分后的排序结果(选第一个记录为枢轴(支点)分割)。

4. 给定关键字序列15, 38, 61, 84, 49, 60, 71, 33, 24, 29, 36，试分别用顺序查找、二分查找、二叉排序树查找、散列查找(其中，散列函数 $H(k)=k \% 11$ ，用线性探查法和拉链法来解决冲突)来实现查找，试画出它们的对应存储形式(顺序查找的顺序表，二分查找的判定树，二叉排序树查找的二叉排序树及两种散列查找的散列表)，并求出每一种查找的成功平均查找长度。

---

5. 某赋权有向图如下图所示。用迪杰斯特拉 (Dijkstra) 算法思想，求源点A到各其余顶点的最短路径及路径长度



(第三大题第5题图)

#### 四、算法设计题（每题10分，共20分）

1. 二叉树以二叉链表存储，设计算法求二叉树中度为2的结点的个数。

2. 假设两个带头结点的链表head和list中的结点值都是整数，且按结点值的递增次序链接起来的链表。在各链表中，每个结点的值各不相同，但head和list可能有值相同的结点（表头结点除外）。编写算法将链表head合并到链表list中，使得合并后的链表是按结点值递增次序链接起来的带头结点的链表，且链表中各个结点的值各不相同。

# 《数据结构》试卷A 参考答案与评分标准

## 一、单项选择（每小题2分，共20分）

1. D	2. B	3. B	4. A	5. B	6. A	7. B	8. C	9. D	10. A
------	------	------	------	------	------	------	------	------	-------

## 二、填空题（每空2分，共20分）

1. 物理位置相邻      2. 中序遍历  
3.

4. 4次

5. 孩子兄弟表示法

6. 移动

7. 健壮性      8. 12534768910; 12345678910      9. n+1

## 三、应用题(第2题6分，1、3、4题每题8分，5题10分，共40分)

1. 解：设A为最小生成树顶点集，B为最小边集，W为权值。求最小路径如下：

- (1) A={0, 1} B={<0, 1>} W=6;  
(2) A={0, 1, 6} B={<0, 1><1, 6>} W=10  
(3) A={0, 1, 6, 2} B={<0, 1><1, 6><6, 2>} W=19  
(4) A={0, 1, 6, 2, 3} B={<0, 1><1, 6><6, 2><2, 3>} W=24  
(5) A={0, 1, 6, 2, 3, 4} B={<0, 1><1, 6><6, 2><2, 3><3, 4>} W=34  
(6) A={0, 1, 6, 2, 3, 4, 5} B={<0, 1><1, 6><6, 2><2, 3><3, 4><0, 5>} W=46

此题评分标准：六个步骤每步1分，画出最小生成树2分，具体情况酌情给分

2. 森林转化为二叉树的过程如下：

此题评分标准：每正确转换一棵二叉树给1分，直接写出结果不扣分，具体情况酌情给分。

3. 快速排序的每趟结果如下：

第一趟排序结果：{12 9 5} 23 {68 39 62 48 33}

第二趟排序结果：{5 9} 12 33 {33 39 62 48} 68

第三趟排序结果：5 9 12 23 33 {39 62 48} 68

第四趟排序结果：5 9 12 23 33 39 {62 48} 68

第五趟排序结果：5 9 12 23 33 39 48 62 68

此题评分标准：正确给出每趟排序结果给2分，排序思想不对，不给分。具体情况酌情给分。

4. 顺序查找

顺序表如下：

存储地址	0	1	2	3	4	5	6	7	8	9	10
数据元素	10	24	32	17	31	30	46	47	40	63	49
查找次数	1	2	3	4	5	6	7	8	9	10	11

---

查找成功平均查找长度ASL=  $(11+1)/2=6$   
二分查找  
二分查找的判定树如下

查找成功平均查找长度ASL=  $(1*1+2*2+3*4+4*4)/11=3$

二叉排序树查找  
二叉排序树如下图所示  
查找成功平均查找长度：

$$\text{ASL} = (1+2+2*3+2*4+3*5+6+7)/11 \\ =45/11 \approx 4$$

10, 24, 32, 17, 31, 30, 46, 47, 40, 63, 49

散列查找：

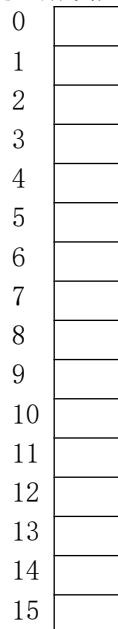
(a) 线性探测法解决冲突得到的散列表如下：

哈希地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
数据元素	32	17	46	47	63	49			24	40	10				30	31
查找次数	1	1	5	5	6	5			1	2	1				1	1

查找成功平均查找长度：

$$\text{ASL} = (1*6+2*5*3+6)/11 = 29/11 = 2.6$$

(b) 拉链法解决冲突得到的散列表如下



查找成功平均查找长度：

$$\text{ASL} = (1*6+2*4+3)/11 = 17/11 = 1.5$$

评分标准：

正确给出顺序查找的存储形式和平均查找长度 ASL 给 2 分

正确给出二分查找的存储形式和平均查找长度 ASL 给 2 分

正确写出二叉排序树查找的存储形式和平均查找长度 ASL 给 2 分

正确写出散列查找（线性探测法）的存储形式和平均长度 ASL 给 2 分

正确写出散列查找（拉链法）的存储形式和平均查找长度 ASL 给 2 分

5.

解：各路径如下：

始点	终点	最短路径	路径长度	评分标准
A	B	( A , C, E, G, B )	1 4	2分
	C	( A C )	2	1分
	D	( A , C , F , D )	1 1	2分
	E	( A , C , E )	1 0	1分
	F	( A , C , F )	6	1分
	G	( A , C , E , G )	1 2	1分

四

1. 判定一棵二叉树是否是完全二叉树，可以使用队列，在遍历中利用完全二叉树“若某结点无左子女就不应有右子女”的原则进行判断。

```
int JudgeComplete(BiTree bt)
{
    tag=0; p=bt; if(p==null) return (1);
    InitQueue(Q); // Q是队列, 初始化队列
    EnQueue(Q,p); // 入队操作, 队中元素是二叉树结点指针, 根结点指针入队
    while (!QueueEmpty(Q)) { //队列非空
        p=DeQueue(Q); //出队
        if (p->lchild && !tag) EnQueue(Q,p->lchild); //左子女入队
        else {
            if (p->lchild) return 0; //前边已有结点为空, 本结点不空
            else tag=1; //首次出现结点为空
        }
        if (p->rchild && !tag) EnQueue(Q,p->rchild); //右子女入队
        else {if (p->rchild) return 0; else tag=1; }
    } //endwhile
    return 1; //该二叉树为完全二叉树
} //JudgeComplete
```

本题答案不唯一，完全二叉树证明还有其它方法。。

正确给出二叉树为空时也为完全二叉树的给2分

正确给出证明二叉树为完全二叉树设计方法的给5分

正确给出队列及相关操作的给3分

若算法设计思想是基于“证明其左子树和右子数均为完全二叉树，由此推出整棵二叉树必是完全二叉树的这一错误结论”的，不给分。

算法思路不完全正确或不完整的，酌情给分或扣分。

2. 本题中的链表有头结点，将表LA分解成表LA和表LB，均带头结点。分解后的LA表含有原表中序号为奇数的元素，LB表含有原A表中序号为偶数的元素。由于要求分解后两表中元素递增有序，故采用在链表尾插入比较方便，使用一指向表尾的指针即可方便实现。

```
void Desassemble(LinkedList LA)
//LA是带头结点的单链表，本算法将其分解成两个带头结点的单链表LA和LB，其中LA表中含原表中序号为奇数的结点，LB表中含原表中序号为偶数的结点。链表中结点的相对顺序同原链表，仍保持递增有序。
{
    i=0;// 整型i记录链表中结点的序号。
    LB=(LinkedList)malloc(sizeof(LNode)); // 创建LB表表头。
    LB->next=null; // LB表的初始化。
    LinkedList ra,rb; // ra和rb将分别指向将创建的LA表和LB表的尾结点。
    ra=LA;rb=LB;
    p=LA->next; // p为链表工作指针，指向待分解的结点。
    LA->next=null; // 置空新的LA表
    while(p) {
        r=p->next; // 暂存p的后继。
        i++; // 结点序号增1
        if(i%2==0) { // 处理原序号为偶数的链表结点。
            p->next=rb->next; // 在LB表尾插入新结点;
    }
```

---

```
rb->next=p; rb=p; // rb指向LB表新的尾结点;
}
else { // 处理原序号为奇数的结点。
    p->next=ra->next; // 在LA表尾插入新结点;
    ra->next=p; ra=p; // ra指向LA表新的尾结点;
}
p=r;           // 将p指向新的待处理结点。
} //EndWhile
```

} // 算法结束

评分标准：

正确给出LA和LB表的初始化操作给2分

正确给出判断链表是否遍历完毕的循环条件给2分

正确分解出符合要求LA表，要点是链表的插入操作，给3分

正确分解出符合要求LB表，要点是链表的插入操作，给3分

算法总体设计正确，但未利用链表LA中的原结点空间，使用辅助结点空间的，扣3分

算法总体设计正确，但分解后的两表中未保持递增有序的，扣3分

算法思路不完全正确或不完整，可酌情给分。