**数据结构试卷（四）**

**一、选择题(每题1分共 20分)**

1．设一维数组中有n个数组元素，则读取第i个数组元素的平均时间复杂度为（ ）。

(A) O(n) (B) O(nlog2n) (C) O(1) (D) O(n2)

2．设一棵二叉树的深度为k，则该二叉树中最多有（ ）个结点。

(A) 2k-1 (B) 2k (C) 2k-1 (D) 2k-1

3．设某无向图中有n个顶点e条边，则该无向图中所有顶点的入度之和为（ ）。

(A) n (B) e (C) 2n (D) 2e

4．在二叉排序树中插入一个结点的时间复杂度为（ ）。

(A) O(1) (B) O(n) (C) O(log2n) (D) O(n2)

5．设某有向图的邻接表中有n个表头结点和m个表结点，则该图中有（ ）条有向边。

(A) n (B) n-1 (C) m (D) m-1

6．设一组初始记录关键字序列为(345，253，674，924，627)，则用基数排序需要进行（ ）趟的分配和回收才能使得初始关键字序列变成有序序列。

(A) 3 (B) 4 (C) 5 (D) 8

7．设用链表作为栈的存储结构则退栈操作（ ）。

(A) 必须判别栈是否为满 (B) 必须判别栈是否为空

(C) 判别栈元素的类型 (D) 对栈不作任何判别

8．下列四种排序中（ ）的空间复杂度最大。

(A) 快速排序 (B) 冒泡排序 (C) 希尔排序 (D) 堆

9．设某二叉树中度数为0的结点数为N0，度数为1的结点数为Nl，度数为2的结点数为N2，则下列等式成立的是（ ）。

(A) N0=N1+1 (B) N0=Nl+N2 (C) N0=N2+1 (D) N0=2N1+l

10.设有序顺序表中有n个数据元素，则利用二分查找法查找数据元素X的最多比较次数不超过（ ）。

(A) log2n+1 (B) log2n-1 (C) log2n (D) log2(n+1)

**二、填空题(每空1分共 20分)**

* 1. 设有n个无序的记录关键字，则直接插入排序的时间复杂度为\_\_\_\_\_\_\_\_，快速排序的平均时间复杂度为\_\_\_\_\_\_\_\_\_。
  2. 设指针变量p指向双向循环链表中的结点X，则删除结点X需要执行的语句序列为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（设结点中的两个指针域分别为llink和rlink）。
  3. 根据初始关键字序列(19，22，01，38，10)建立的二叉排序树的高度为\_\_\_\_\_\_\_\_\_\_\_\_。
  4. 深度为k的完全二叉树中最少有\_\_\_\_\_\_\_\_\_\_\_\_个结点。
  5. 设初始记录关键字序列为(K1，K2，…，Kn)，则用筛选法思想建堆必须从第\_\_\_\_\_\_个元素开始进行筛选。
  6. 设哈夫曼树中共有99个结点，则该树中有\_\_\_\_\_\_\_\_\_个叶子结点；若采用二叉链表作为存储结构，则该树中有\_\_\_\_\_个空指针域。
  7. 设有一个顺序循环队列中有M个存储单元，则该循环队列中最多能够存储\_\_\_\_\_\_\_\_个队列元素；当前实际存储\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_个队列元素（设头指针F指向当前队头元素的前一个位置，尾指针指向当前队尾元素的位置）。
  8. 设顺序线性表中有n个数据元素，则第i个位置上插入一个数据元素需要移动表中\_\_\_\_\_\_\_个数据元素；删除第i个位置上的数据元素需要移动表中\_\_\_\_\_\_\_个元素。
  9. 设一组初始记录关键字序列为(20，18，22，16，30，19)，则以20为中轴的一趟快速排序结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
  10. 设一组初始记录关键字序列为(20，18，22，16，30，19)，则根据这些初始关键字序列建成的初始堆为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
  11. 设某无向图G中有n个顶点，用邻接矩阵A作为该图的存储结构，则顶点i和顶点j互为邻接点的条件是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
  12. 设无向图对应的邻接矩阵为A，则A中第i上非0元素的个数\_\_\_\_\_\_\_\_\_第i列上非0元素的个数（填等于，大于或小于）。
  13. 设前序遍历某二叉树的序列为ABCD，中序遍历该二叉树的序列为BADC，则后序遍历该二叉树的序列为\_\_\_\_\_\_\_\_\_\_\_\_\_。
  14. 设散列函数H(k)=k mod p，解决冲突的方法为链地址法。要求在下列算法划线处填上正确的语句完成在散列表hashtalbe中查找关键字值等于k的结点，成功时返回指向关键字的指针，不成功时返回标志0。

typedef struct node {int key; struct node \*next;} lklist;

void createlkhash(lklist \*hashtable[ ])

{

int i,k; lklist \*s;

for(i=0;i<m;i++)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

for(i=0;i<n;i++)

{

s=(lklist \*)malloc(sizeof(lklist)); s->key=a[i];

k=a[i] % p; s->next=hashtable[k];\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

}

**三、计算题(每题10分，共30分)**

1、画出广义表LS=(( ) , (e) , (a , (b , c , d )))的头尾链表存储结构。

2、下图所示的森林：

(1) 求树（a）的先根序列和后根序列；

(2) 求森林先序序列和中序序列；

（3）将此森林转换为相应的二叉树；



3、设散列表的地址范围是[ 0..9 ]，散列函数为H（key）= （key 2 +2）MOD 9,并采用链表处理冲突，请画出元素7、4、5、3、6、2、8、9依次插入散列表的存储结构。

**四、算法设计题(每题10分，共30分)**

* + 1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。
  1. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。
  2. 在链式存储结构上建立一棵二叉排序树。

**一、选择题**

1．C 2．D 3．D 4．B 5．C

6．A 7．B 8．A 9．C 10．A

**二、填空题**

1. O(n2)，O(nlog2n)
2. p>llink->rlink=p->rlink; p->rlink->llink=p->rlink
3. 3
4. 2k-1
5. n/2
6. 50，51
7. m-1，(R-F+M)%M
8. n+1-i，n-i
9. (19，18，16，20，30，22)
10. (16，18，19，20，32，22)
11. A[i][j]=1
12. 等于
13. BDCA
14. hashtable[i]=0，hashtable[k]=s

**三、计算题**

**1．**



2．

(1) ABCDEF; BDEFCA；(2) ABCDEFGHIJK; BDEFCAIJKHG林转换为相应的二叉树；



3．H(4)=H(5)=0,H(3)=H(6)=H(9)=2,H(8)=3,H(2)=H(7)=6



**四、算法设计题**

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。

typedef char datatype;

typedef struct node {datatype data; struct node \*next;}lklist;

void split(lklist \*head,lklist \*&ha,lklist \*&hb,lklist \*&hc)

{

lklist \*p; ha=0,hb=0,hc=0;

for(p=head;p!=0;p=head)

{

head=p->next; p->next=0;

if (p->data>='A' && p->data<='Z') {p->next=ha; ha=p;}

else if (p->data>='0' && p->data<='9') {p->next=hb; hb=p;} else {p->next=hc; hc=p;}

}

}

1. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。

typedef struct node {int data; struct node \*lchild,\*rchild;} bitree;

void swapbitree(bitree \*bt)

{

bitree \*p;

if(bt==0) return;

swapbitree(bt->lchild); swapbitree(bt->rchild);

p=bt->lchild; bt->lchild=bt->rchild; bt->rchild=p;

}

1. 在链式存储结构上建立一棵二叉排序树。

#define n 10

typedef struct node {

int key;

struct node \*lchild,\*rchild;

} bitree;

void bstinsert(bitree \*&bt,int key) {

if (bt==0) {

bt=(bitree \*)malloc(sizeof(bitree));

bt->key=key;

bt->lchild=bt->rchild=0;

} else if (bt->key>key)

bstinsert(bt->lchild,key);

else

bstinsert(bt->rchild,key);

}

void createbsttree(bitree \*&bt) {

int i;

for(i=1; i<=n; i++)

bstinsert(bt,random(100));

}