**数据结构试卷（三）**

**一、选择题(30分)**

1．设某数据结构的二元组形式表示为A=(D，R)，D={01，02，03，04，05，06，07，08，09}，R={r}，r={<01，02>，<01，03>，<01，04>，<02，05>，<02，06>，<03，07>，<03，08>，<03，09>}，则数据结构A是（ ）。

(A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构

2．下面程序的时间复杂为（ ）

for（i=1，s=0； i<=n； i++） {t=1；for(j=1；j<=i；j++) t=t\*j；s=s+t；}

(A) O(n) (B) O(n2) (C) O(n3) (D) O(n4)

3．设指针变量p指向单链表中结点A，若删除单链表中结点A，则需要修改指针的操作序列为（ ）。

(A) q=p->next；p->data=q->data；p->next=q->next；free(q)；

(B) q=p->next；q->data=p->data；p->next=q->next；free(q)；

(C) q=p->next；p->next=q->next；free(q)；

(D) q=p->next；p->data=q->data；free(q)；

4．设有n个待排序的记录关键字，则在堆排序中需要（ ）个辅助记录单元。

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

5．设一组初始关键字记录关键字为(20，15，14，18，21，36，40，10)，则以20为基准记录的一趟快速排序结束后的结果为( )。

(A) 10，15，14，18，20，36，40，21

(B) 10，15，14，18，20，40，36，21

(C) 10，15，14，20，18，40，36，2l

(D) 15，10，14，18，20，36，40，21

6．设二叉排序树中有n个结点，则在二叉排序树的平均平均查找长度为（ ）。

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

7．设无向图G中有n个顶点e条边，则其对应的邻接表中的表头结点和表结点的个数分别为（ ）。

(A) n，e (B) e，n (C) 2n，e (D) n，2e

8. 设某强连通图中有n个顶点，则该强连通图中至少有（ ）条边。

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

9．设有5000个待排序的记录关键字，如果需要用最快的方法选出其中最小的10个记录关键字，则用下列（ ）方法可以达到此目的。

(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 插入排序

10.下列四种排序中（ ）的空间复杂度最大。

(A) 插入排序 (B) 冒泡排序 (C) 堆排序 (D) 归并排序

**二、填空殖(48分，其中最后两小题各6分)**

1. 1.         数据的物理结构主要包括\_\_\_\_\_\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_两种情况。
2. 2.         设一棵完全二叉树中有500个结点，则该二叉树的深度为\_\_\_\_\_\_\_\_\_\_；若用二叉链表作为该完全二叉树的存储结构，则共有\_\_\_\_\_\_\_\_\_\_\_个空指针域。
3. 3.         设输入序列为1、2、3，则经过栈的作用后可以得到\_\_\_\_\_\_\_\_\_\_\_种不同的输出序列。
4. 4.         设有向图G用邻接矩阵A[n][n]作为存储结构，则该邻接矩阵中第i行上所有元素之和等于顶点i的\_\_\_\_\_\_\_\_，第i列上所有元素之和等于顶点i的\_\_\_\_\_\_\_\_。
5. 5.         设哈夫曼树中共有n个结点，则该哈夫曼树中有\_\_\_\_\_\_\_\_个度数为1的结点。
6. 6.         设有向图G中有n个顶点e条有向边，所有的顶点入度数之和为d，则e和d的关系为\_\_\_\_\_\_\_\_\_。
7. 7.         \_\_\_\_\_\_\_\_\_\_遍历二叉排序树中的结点可以得到一个递增的关键字序列（填先序、中序或后序）。
8. 8.         设查找表中有100个元素，如果用二分法查找方法查找数据元素X，则最多需要比较\_\_\_\_\_\_\_\_次就可以断定数据元素X是否在查找表中。
9. 9.         不论是顺序存储结构的栈还是链式存储结构的栈，其入栈和出栈操作的时间复杂度均为\_\_\_\_\_\_\_\_\_\_\_\_。
10. 10.     设有n个结点的完全二叉树，如果按照从自上到下、从左到右从1开始顺序编号，则第i个结点的双亲结点编号为\_\_\_\_\_\_\_\_\_\_\_\_，右孩子结点的编号为\_\_\_\_\_\_\_\_\_\_\_。
11. 11.     设一组初始记录关键字为(72，73，71，23，94，16，5)，则以记录关键字72为基准的一趟快速排序结果为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
12. 12.     设有向图G中有向边的集合E={<1，2>，<2，3>，<1，4>，<4，2>，<4，3>}，则该图的一种拓扑序列为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
13. 13.     下列算法实现在顺序散列表中查找值为x的关键字，请在下划线处填上正确的语句。

struct record{int key; int others;};

int hashsqsearch(struct record hashtable[ ],int k)

{

int i,j; j=i=k % p;

while (hashtable[j].key!=k&&hashtable[j].flag!=0){j=(\_\_\_\_) %m; if (i==j) return(-1);}

if (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ) return(j); else return(-1);

}

1. 14.     下列算法实现在二叉排序树上查找关键值k，请在下划线处填上正确的语句。

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

bitree \*bstsearch(bitree \*t, int k)

{

if (t==0 ) return(0);else while (t!=0)

if (t->key==k)\_\_\_\_\_\_\_\_\_\_\_\_\_; else if (t->key>k) t=t->lchild; else\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

**三、算法设计题(22分)**

1. 1．  设计在单链表中删除值相同的多余结点的算法。
2. 2．  设计一个求结点x在二叉树中的双亲结点算法。

**数据结构试卷（三）参考答案**

**一、选择题**

1.B 2.B 3.A 4.A 5.A

6.B 7.D 8.C 9.B 10.D

第3小题分析：首先用指针变量q指向结点A的后继结点B，然后将结点B的值复制到结点A中，最后删除结点B。

第9小题分析：9快速排序、归并排序和插入排序必须等到整个排序结束后才能够求出最小的10个数，而堆排序只需要在初始堆的基础上再进行10次筛选即可，每次筛选的时间复杂度为O(log2n)。

**二、填空题**

1. 1.        顺序存储结构、链式存储结构
2. 2.        9，501
3. 3.        5
4. 4.        出度，入度
5. 5.        0
6. 6.        e=d
7. 7.        中序
8. 8.        7
9. 9.        O(1)
10. 10.     i/2，2i+1
11. 11.     (5，16，71，23，72，94，73)
12. 12.     (1，4，3，2)
13. 13.     j+1，hashtable[j].key==k
14. 14.     return(t)，t=t->rchild

第8小题分析：二分查找的过程可以用一棵二叉树来描述，该二叉树称为二叉判定树。在有序表上进行二分查找时的查找长度不超过二叉判定树的高度1+log2n。

**三、算法设计题**

1. 1.        设计在单链表中删除值相同的多余结点的算法。

typedef int datatype;

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

void delredundant(lklist \*&head)

{

lklist \*p,\*q,\*s;

for(p=head;p!=0;p=p->next)

{

for(q=p->next,s=q;q!=0; )

if (q->data==p->data) {s->next=q->next; free(q);q=s->next;}

else {s=q,q=q->next;}

}

}

1. 2.        设计一个求结点x在二叉树中的双亲结点算法。

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

bitree \*q[20]; int r=0,f=0,flag=0;

void preorder(bitree \*bt, char x)

{

if (bt!=0 && flag==0)

if (bt->data==x) { flag=1; return;}

else {r=(r+1)% 20; q[r]=bt; preorder(bt->lchild,x); preorder(bt->rchild,x); }

}

void parent(bitree \*bt,char x)

{

int i;

preorder(bt,x);

for(i=f+1; i<=r; i++) if (q[i]->lchild->data==x || q[i]->rchild->data) break;

if (flag==0) printf("not found x\n");

else if (i<=r) printf("%c",bt->data); else printf("not parent");

}