**数据结构试卷（五）**

**一、选择题(20分)**

1．数据的最小单位是（ ）。

(A) 数据项 (B) 数据类型 (C) 数据元素 (D) 数据变量

2．设一组初始记录关键字序列为(50，40，95，20，15，70，60，45)，则以增量d=4的一趟希尔排序结束后前4条记录关键字为（ ）。

(A) 40，50，20，95 (B) 15，40，60，20

(C) 15，20，40，45 (D) 45，40，15，20

3．设一组初始记录关键字序列为(25，50，15，35，80，85，20，40，36，70)，其中含有5个长度为2的有序子表，则用归并排序的方法对该记录关键字序列进行一趟归并后的结果为（ ）。

(A) 15，25，35，50，20，40，80，85，36，70

(B) 15，25，35，50，80，20，85，40，70，36

(C) 15，25，35，50，80，85，20，36，40，70

(D) 15，25，35，50，80，20，36，40，70，85

4．函数substr(“DATASTRUCTURE”，5，9)的返回值为（ ）。

(A) “STRUCTURE” (B) “DATA”

(C) “ASTRUCTUR” (D) “DATASTRUCTURE”

5．设一个有序的单链表中有n个结点，现要求插入一个新结点后使得单链表仍然保持有序，则该操作的时间复杂度为（ ）。

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

6．设一棵m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m的结点数为Nm，则N0=（ ）。

(A) Nl+N2+……+Nm (B) l+N2+2N3+3N4+……+(m-1)Nm

(C) N2+2N3+3N4+……+(m-1)Nm (D) 2Nl+3N2+……+(m+1)Nm

7．设有序表中有1000个元素，则用二分查找查找元素X最多需要比较（ ）次。

(A) 25 (B) 10 (C) 7 (D) 1

8．设连通图G中的边集E={(a，b)，(a，e)，(a，c)，(b，e)，(e，d)，(d，f)，(f，c)}，则从顶点a出发可以得到一种深度优先遍历的顶点序列为（ ）。

(A) abedfc (B) acfebd (C) aebdfc (D) aedfcb

9．设输入序列是1、2、3、……、n，经过栈的作用后输出序列的第一个元素是n，则输出序列中第i个输出元素是（ ）。

(A) n-i (B) n-1-i (C) n+1-i (D) 不能确定

10 设一组初始记录关键字序列为(45，80，55，40，42，85)，则以第一个记录关键字45为基准而得到一趟快速排序的结果是（ ）。

(A) 40，42，45，55，80，83 (B) 42，40，45，80，85，88

(C) 42，40，45，55，80，85 (D) 42，40，45，85，55，80

**二、填空题(共20分)**

1. 设有一个顺序共享栈S[0：n-1]，其中第一个栈项指针top1的初值为-1，第二个栈顶指针top2的初值为n，则判断共享栈满的条件是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
2. 在图的邻接表中用顺序存储结构存储表头结点的优点是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
3. 设有一个n阶的下三角矩阵A，如果按照行的顺序将下三角矩阵中的元素（包括对角线上元素）存放在n(n+1)个连续的存储单元中，则A[i][j]与A[0][0]之间有\_\_\_\_\_\_\_个数据元素。
4. 栈的插入和删除只能在栈的栈顶进行，后进栈的元素必定先出栈，所以又把栈称为\_\_\_\_\_\_\_\_\_\_表；队列的插入和删除运算分别在队列的两端进行，先进队列的元素必定先出队列，所以又把队列称为\_\_\_\_\_\_\_\_\_表。
5. 设一棵完全二叉树的顺序存储结构中存储数据元素为ABCDEF，则该二叉树的前序遍历序列为\_\_\_\_\_\_\_\_\_\_\_，中序遍历序列为\_\_\_\_\_\_\_\_\_\_\_，后序遍历序列为\_\_\_\_\_\_\_\_\_\_\_。
6. 设一棵完全二叉树有128个结点，则该完全二叉树的深度为\_\_\_\_\_\_\_\_，有\_\_\_\_\_\_\_\_\_\_个叶子结点。
7. 设有向图G的存储结构用邻接矩阵A来表示，则A中第i行中所有非零元素个数之和等于顶点i的\_\_\_\_\_\_\_\_，第i列中所有非零元素个数之和等于顶点i的\_\_\_\_\_\_\_\_\_\_。
8. 设一组初始记录关键字序列(k1，k2，……，kn)是堆，则对i=1，2，…，n/2而言满足的条件为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
9. 下面程序段的功能是实现冒泡排序算法，请在下划线处填上正确的语句。

void bubble(int r[n])

{

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

{

for(exchange=0,j=0; j<\_\_\_\_\_\_\_\_\_\_\_\_\_;j++)

if (r[j]>r[j+1]){temp=r[j+1];\_\_\_\_\_\_\_\_\_\_\_\_\_\_;r[j]=temp;exchange=1;}

if (exchange==0) return；

}

}

1. 下面程序段的功能是实现二分查找算法，请在下划线处填上正确的语句。

struct record{int key; int others;};

int bisearch(struct record r[ ], int k)

{

int low=0,mid,high=n-1;

while(low<=high)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

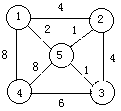
if(r[mid].key==k) return(mid+1); else if(\_\_\_\_\_\_\_\_\_\_\_\_) high=mid-1;else low=mid+1;

}

return(0);

}

**三、应用题(32分)**

1. 设某棵二叉树的中序遍历序列为DBEAC，前序遍历序列为ABDEC，要求给出该二叉树的的后序遍历序列。
2. 设无向图G（如右图所示），给出该图的最小生成树上边的集合并计算最小生成树各边上的权值之和。
3. 设一组初始记录关键字序列为(15，17，18，22，35，51，60)，要求计算出成功查找时的平均查找长度。
4. 设散列表的长度为8，散列函数H(k)=k mod 7，初始记录关键字序列为(25，31，8，27，13，68)，要求分别计算出用线性探测法和链地址法作为解决冲突方法的平均查找长度。

**四、算法设计题(28分)**

1. 设计判断两个二叉树是否相同的算法。
2. 设计两个有序单链表的合并排序算法。

**数据结构试卷（五）参考答案**

**一、选择题**

1．A 2．B 3．A 4．A 5．D

6．B 7．B 8．B 9．C 10．C

**二、填空题**

1. top1+1=top2
2. 可以随机访问到任一个顶点的简单链表
3. i(i+1)/2+j-1
4. FILO，FIFO
5. ABDECF，DBEAFC，DEBFCA
6. 8，64
7. 出度，入度
8. ki<=k2i && ki<=k2i+1
9. n-i，r[j+1]=r[j]
10. mid=(low+high)/2，r[mid].key>k

**三、应用题**

1. DEBCA
2. E={(1,5),(5,2),(5,3),(3,4)},W=10
3. ASL=(1\*1+2\*2+3\*4)/7=17/7
4. ASL1=7/6，ASL2=4/3

**四、算法设计题**

1. 设计判断两个二叉树是否相同的算法。

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

int judgebitree(bitree \*bt1,bitree \*bt2)

{

if (bt1==0 && bt2==0) return(1);

else if (bt1==0 || bt2==0 ||bt1->data!=bt2->data) return(0);

else return(judgebitree(bt1->lchild,bt2->lchild)\*judgebitree(bt1->rchild,bt2->rchild));

}

1. 设计两个有序单链表的合并排序算法。

void mergelklist(lklist \*ha,lklist \*hb,lklist \*&hc)

{

lklist \*s=hc=0;

while(ha!=0 && hb!=0)

if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}

else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}

if(ha==0) s->next=hb; else s->next=ha;

}