**数据结构 试卷**

一、选择题（每题1分）   
1．循环队列用数组A[0，m-1]存放其元素值，已知其头尾指针分别是front和rear，则当前队列中的元素个数是（）。   
  
A．(rear-front+m)%m    B．read-front+1    C．read-front-1     D．read-front   
  
2．广义表((a,b),c,d)的表头是（ ），表尾是（ ）。   
  
A．a      B．b       C．(a,b)       D．(c,d)   
  
3．将递归算法转换成对应的非递归算法时，通常需要使用（ ）。   
  
A．栈   B．队列       C．链表         D．树   
  
4．某二叉树的前序遍历结点访问顺序是abdgcefh，中序遍历的结点访问顺序是dgbaechf，则其后序遍历的结点访问顺序是（ ）。   
  
A．bdgcefha      B．gdbecfha         C．bdgaechf        D．gdbehfca   
  
5．按照二叉树的定义，具有3个结点的二叉树有( )种。   
  
A．3    B．4        C．5        D．6   
  
6．深度为5的二叉树至多有( )个结点。   
  
A．10      B．16        C．31       D．32   
  
7．在一个具有n个顶点的无向图中，要连通全部顶点至少需要（ ）条边。   
  
A．n      B．n+1      C．n-1       D．n/2   
  
8．有一个有序表为{1,3,9,12,32,41,45,62,75,77,82,95,100}，当二分查找值为82的结点时，（ ）次比较后查找成功。   
  
A．1 B．2 C．4 D．8   
  
9．在待排序的元素序列基本有序的前提下，效率最高的排序方法是（ ）。   
  
A．插入排序 B．选择排序 C．快速排序 D．归并排序

10．用某种排序方法对线性表（25,84,21,47,15,27,68,35,20）进行排序时，元素序列的变化情况如下：   
  
25,84,21,47,15,27,68,35,20   
  
15,84,21,47,25,27,68,35,20   
  
15,20,21,47,25,27,68,35,84   
  
15,20,21,25,47,27,68,35,84   
．   
．   
15,20,21,25,27,35,47,68,84   
则采用的排序方法是（ ）。   
  
A．选择排序 B．快速排序 C．归并排序 D．冒泡排序   
  
二、填空题（每题2分）   
1．在一个稀疏矩阵中，每个非零元素所对应的三元组包括该元素的\_\_\_\_、\_\_\_\_和\_\_\_\_\_\_\_ 三项。   
  
2．在一棵三叉树中，度为3的结点数为2个，度为2的结点数有1个，度为1的结点数为2个，那么度为0的结点数有\_\_\_\_\_\_个。   
  
3．在一个顺序存储的队列Q中，判断队空的条件为\_\_\_\_\_\_\_\_\_\_，判断队满的条件为\_\_\_\_\_\_\_\_\_\_。   
  
4．假定一棵树的广义表表示为A（B（C，D（E，F，G），H（I，J））），则树中所含的结点数为\_\_\_\_\_\_个，树的深度为\_\_\_\_\_，树的度为\_\_\_\_\_\_。   
  
5．一棵二叉树的结点数为18，则它的最小深度为\_\_\_\_\_\_\_，最大深度为\_\_\_\_\_\_\_。   
  
6．对于一棵具有n个结点的二叉树，对应二叉链表中指针总数为\_\_\_\_\_\_个，其中\_\_\_\_\_\_个用于指向孩子结点，\_\_\_\_\_\_个指针空闲着。   
  
7．假定对线性表（38，25，74，52，48）进行散列存储，采用H（K）=K%7作为散列函数，若分别采用线性探测法和链接法处理冲突，则对各自散列表进行查找的平均查找长度分别为\_\_\_\_和\_\_\_\_\_\_。   
  
8．算术表达式a+b/(c-(d-e)) ′f的逆波兰式是\_\_\_\_\_\_\_\_\_\_\_\_\_\_。   
  
9．设二维数组a[10][10] 是对称阵，现将a中的下三角（含对角线）元素以行为主序存储在首地址为2000的存储区域中，每个元素占3个单元，则元素a[7][6]的地址为\_\_\_\_\_\_\_\_。   
  
10．设有一稀疏图G，则G采用\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_存储较省空间。

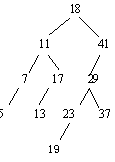
三、求解题   
1．已知序列 18,11,17,7,5,13,41,29,37,23,19。请画出相应的二叉排序树并写出该树的前序、中序和后序序列。（8分）   
  
  
  
  
  
  
  
2．对于线性表F=25,40,20,27,45,50,15,35,17,30,12，按照下面给出的排序方法分别对F进行排序。请分别给出在各遍结束时线性表的状况。（10分）   
(1) 快速排序   
(2) 归并排序   
  
  
  
3．假定无向图G有7个结点和12条边,并依次输入这12条边(1,2),(1,3),(1,5)，(1,6), (1,7)，(2,3),(3,4),(3,5), (3,7)，(4,6)，(5,6)，(5,7).请画出图G的邻接表.并按你所给的邻接表,从结点1出发,分别按深度优先搜索法和广度优先搜索法进行遍历的结点序列. （10分）   
  
  
  
4．已知权值: 11,17,7,5,13,41,29,37,23,19。请画出相应的哈夫曼树。（4分）   
  
  
  
四、读程填空（每个空格处只填一个语句或表达式）（每个空格2分）   
1．完成下列排序算法。   
typedef struct node   
  {int key;   
   int data;   
  }NODE;   
void insert(NODE r[ ],int n)   
{ int i,j;   
 for(i=2;i<=n;i++)   
    if (       A         )   
    {r[0]=r[i];j=i-1;   
     while (       B        )   
     r[j+1]=r[j--];   
          C          ;   
    }   
}   
  
2．完成下列将两个递增有序单链表合并成一个递增有序单链表的算法。   
typedef struct lnode {   
    elemtype data;   
    struct lnode \*next;   
}lnode,\*linklist;   
linklist meger(linklist pa,linklist pb)   
{ p=pa; q=pb;   
  while (       A         )   
     if (       B        ) p=p->next ;   
     if (p->next->data==q->next->data)   
             {u=q->next;        C        ;free(u);}   
     else{u=q->next;q->next=u->next;   
           D        ; p->next=u;         E         ;}   
 if (q->next)          F        ;   
 free(pb);   
 return(pa);   
}   
  
3．完成下列前序遍历二叉树算法。   
#define max 100   
typedef struct tnode{   
int data;   
struct tnode \*lchild,\*rchild;   
}tnode;   
typedef struct stack {   
   tnode \*elem[max];   
   int top;   
}stack;   
void print\_tree(tnode \*bt)   
{stack s; tnode \*p;   
p=\_\_\_\_A\_\_\_\_;s.top=0;   
while (p || s.top)   
  {   while (        B        )   
       { printf("%d",p->data);s.elem[s.top++]=p;      C       ;}   
     if (s.top>0){      D       ;        E         ;}   
  }   
}   
  
4．下面的函数在给定的二叉排序树中插入一个新结点。在插入之前,首先对查找树进行查找,如树中不包含插入的新结点,则进行插入;否则不插入。   
typedef struct tnode {   
char data;   
struct tnode \*lchild,\*rchild;   
}tnode，\*bitreptr;   
bitreprt insert(bitreptr root, char ch)   
{ bitreptr p,q;   
  p=NULL;q=root;   
  while (q && q->data!=ch)   
      { p=q;   
        if( ch<q->data)       A        ;   
        else         B          ;   
      }   
 if (!q) {q=(bitreptr)malloc(sizeof(tnode));   
 q->data=ch;q->lchild=q->rchild=NULL;   
 if (!p)       C        ;   
 else if (ch>p->data)        D       ;   
      else        E         ;   
}   
return root;   
}

**数据结构 试卷答案**

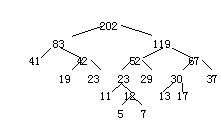
 一、 选择题   
1.A 2.C D 3.A 4.D 5.C 6.C 7.C 8.C 9.A 10.A   
  
二、 填空题   
1. 行、列、元素值      2. 6        3. front==rear front==(rear+1)%size     4. 10 4 3      5. 5

6.2n n-1 n+1           7. 2 1.4    8. abcde--/f\*+              9.2102         10.邻接表

三、求解题   
1．



前序：18 11 7 5 17 13 41 29 23 19 37   
中序：5 7 11 13 17 18 19 23 29 37 41   
后序：5 7 13 17 11 19 23 37 29 41 18   
  
2. 快速排序：   
25 40 20 27 45 50 15 35 17 30 12   
12 17 20 15 25 50 45 35 27 30 40   
12 17 20 15 25 40 45 35 27 30 50   
12 15 17 20 25 30 27 35 40 45 50   
12 15 17 20 25 27 30 35 40 45 50   
  
归并排序：   
25 40 20 27 45 50 15 35 17 30 12   
25 40 20 27 45 50 15 35 17 30 12   
20 25 27 40 15 35 45 50 12 17 30   
15 20 25 27 35 40 45 50 12 17 30   
12 15 17 20 25 27 30 35 40 45 50   
3.   
1-〉2 3 5 6 7   
2-〉1 3   
3-〉1 2 4 5 7   
4-〉3 6   
5-〉1 3 6 7   
6–〉1 4 5   
7–〉1 3 5   
  
DFS:1 2 3 4 6 5 7   
BFS:1 2 3 5 6 7 4

4.  


四、程序填空题   
1. A. r[i].key<r[i-1].key B. r[0].key<r[j].key C.r[j+1]=r[0]   
  
2. A. p->next&&q->next   B.p->next->data<q->next->data C.q->next=u->next   
   D. u->next=p->next    E. p=u      F.p->next=q->next   
  
3. A.bt        B.p       C.p=p->lchild        D.p=s.elem[--s.top]     E.p=p->rchild   
   
4. A. q=q->lchild     B.q=q->rchild     C.root=q     D.p->rchild=q    E. p->lchild=q