杭 州 电 子 科 技 大 学

2006 年攻读硕士学位研究生入学考试

《数据结构》试题

(试卷共五大题，5 页，150 分)

【所有答案必须写在答题纸上，做在试卷或草稿纸上无效!】

一．是非题（每题 2 分，共 20 分）

（正确的用 T 表示，错误的用 F 表示。）

1. 抽象数据类型可用三元式（D，S，P）表示。其中：D 是数据对象，S 是 D 上的关系，P 是对 D 的基本操作集。
2. 链式存储方式的优点是对数据元素的插入、删除无需移动表中的数据元素。
3. 带头结点的单链表和不带头结点的单链表对首元结点的操作没有区别。
4. 满二叉树不是完全二叉树。
5. 已知某二叉树的中序和后序遍历序列未必能得到该二叉树。
6. 强连通分量包含了强连通部分的所有顶点及足以构成连通的若干条弧。
7. 在伙伴系统中，已知某地址 P mod 2^{k+1}=0,其伙伴块为 p-2^k。
8. 栈和队列是操作上受限制的线性表。
9. 对 n 个数据的排序所能达到的最好时间复杂度为 O(n^2) 。
10. 二叉排序树的查找长度至多为 $\log_2 n$ 。

二．填空题（每空 2 分，共 30 分）

1. 若对编号为 1，2，3，的列车车厢依次通过扳道栈进行调度，不能得到（　　）的序列。
2. 设二维数组 A[m][n]中每一数组元素占 k 个存储单元，若以行序为主序进行存储，则 A(i,j)的地址 LOC(i,j)为（　　）。
3. 设 LS=（$a_1, a_2, \ldots\ldots a_n$ ）是一广义表，其中（　　）是 LS 的表头，（　　）是 LS 的表尾。
4. 设森林 F 中有三棵树，第一、第二和第三棵树的结点个数分别为 m1、m2 和 m3，则与森林 F 对应的二叉树根结点的右子树上的结点个数是（　　）。
5. 设哈夫曼(Huffman)树的高度为 h(h>1)，则该哈夫曼树中所包含的树叶结点数至少是（　　），至多是（　　）。
6. 对 n 个数据的查找所能达到的最好时间复杂度为(O(　　))。
7. 一棵 m 阶的 B-树，第一层至少有一个结点；第二层至少有 2 个结点，

除根之外的所有非终端结点至少有（　　）棵子树，树中每个结点至多有（　　）棵子树。

8. 无向图的连通分量是其（　　）的连通子图。

9. 已知一组待排序的记录关键字初始排列如下：

45，67，23，64，12，68，09，34，46，77，25，36．则：

按升序一趟快速排序的结果是（　　）；

按升序一趟希尔排序（初始步长为 3）的结果是（　　）；

按升序一趟基数排序的结果是（　　）；

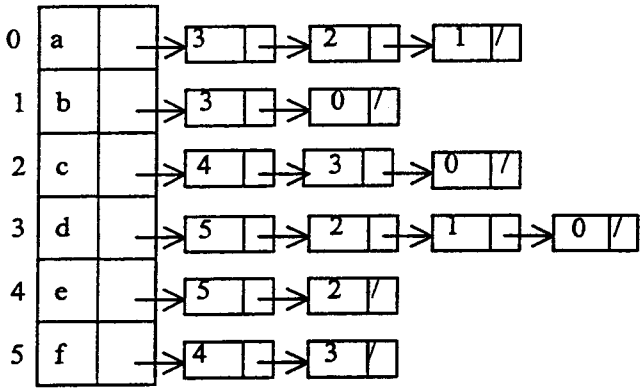
建立初始堆（小顶堆）的结果是（　　）。

三．问答题（每题 8 分，共 40 分）

1. 比较链式栈、顺序栈，链式队列、顺序（循环）队列的优缺点。

2. 已知某森林的先序遍历次序为：ABCDEFGIJKH，中序遍历次序为：BACEFIKJGHD。画出该森林。并画出将其转换为二叉树后的后序线索。

3. 已知某无向图的邻接表如下所示：



（1）画出其原图。

（2）根据邻接表给出深度优先遍历次序及广度优先遍历次序。

（3）画出邻接多重表存储结构。

4. 根据插入次序（80，40，20，60，100，70）建立平衡的二叉排序树。图示结构变化结果。（请给出中间过程）

5. 设哈希函数为 H(key)=key mod 13，一组关键字为：34，56，12，58，06，45，27，91，36，18，55，23。

（1）画出用链地址法处理冲突的哈希表。

（2）画出用公共溢出区法处理冲突的哈希表。

四．算法分析：指出以下算法的功能（每题 6 分，共 24 分）

1．
```
void    B1 (BiTree T,   int& count){
  //已知 T 是二叉树的根结点，count 初值为 0
      if (T) {
          if ((T->lchild)&& (T->rchild))   count++;
          B1(T->lchild, count);
          B1(T->rchild, count);
      } //B1
```

2．
```
void    B2 (SqQueue &Q){
  //已知 Q 是一循环队列（顺序映象）
      if(Q.front==Q.rear)    exit;
      m=(Q.rear-Q.front+ MAXQSIZE)% MAXQSIZE ;
      f=Q.front;
      r=(Q.rear-1+ MAXQSIZE)% MAXQSIZE;
      for (i=1;i<=m/2;i++) {
          Q.base[f]<->Q.base[r];
          f=(f+1)% MAXQSIZE;
          r=(Q.rear-1+ MAXQSIZE)% MAXQSIZE;
      }//for
  }//B2
```

3．
```
void    B3 ( BiTree &p ){
      // p 为二叉排序树中某结点
      if (!p->rchild)   {
          q = p;   p = p->lchild;   free(q);
      }
      else if (!p->lchild) {
          q = p;   p = p->rchild;   free(q);
      }
      else {
          q = p;   s = p->lchild;
          while (!s->rchild) { q = s;   s = s->rchild; }
          p->data = s->data;
          if (q != p )   q->rchild = s->lchild;
          else   q->lchild = s->lchild;
          free(s);
```

```
            }
    } // B3

4. Void    B4 (string &s, string t, string v) {
           //已知 s,t,v 均为字符串
           M = strlength(s);
           N = strlength(t);
           L = strlength(v);
           pos = 1;
           i = index(s,t,pos);
           while(i!=0) {
                substring(sub1,s,1,i-1);
                substring(sub2,s,i+N,M-(i+N)+1);
                concat(s,sub1,v);
                concat(s,s,sub2);
                M=length(s);
                pos=i+L;
                i=index(s,t,pos);
           }//while
    }//B4
```

五．算法设计（第题 12 分，共 36 分）

1．已知某线性表以带头结点的单链表表示，结点结构为：

```
Typedef struct   LNode {
        ElemType        data;
        struct Lnode    *next;
    } LNode, *LinkList;
```

写一算法 Oplinklist(linklist L,int i;int n;int j)

删除线性表 a_i至a_{i+n-1}共 n 个元素,并将之插入至原表中的第 j 个元素之前。(其中：j<i　or　j>=i+n)

2．已知树的存储结构为：

```
typedef struct CSNode{
    Elem    data;
    struct CSNode    *firstchild,   *nextsibling;
} CSNode, *CSTree;
```

写一求树的深度的算法 int TreeDepth(CSTree T)

4

3．已知无向图邻接多重表存储结构为：

```
typedef struct Ebox {
    int     ivex, jvex;
    struct EBox   *ilink, *jlink;
    InfoType        *info;
} EBox;
typedef struct VexBox {
    VertexType   data;
    EBox   *firstedge;
} VexBox;
typedef struct {
    VexBox   adjmulist[MAX_VERTEX_NUM];
    int     vexnum, edgenum;
} AMLGraph
```

写一建立图的邻接多重表存储结构的函数：

CreateUDG(ANLGraph &G)