北京师范大学
2003 年招收硕士学位研究生入学考试试题

专　　业：计算机应用技术　　科目代码：496

研究方向：网络应用与信息安全、人工智能　　考试科目：计算机基础

图形学与人机交互、计算机教育应用

注：全部试题请答在答题纸上，标清题号即可。

一、请翻译成中文　（12 分）：

1. allocation strategy
2. boundary tag method
3. merge insertion sort
4. pattern matching
5. threaded binary trees
6. adjacency multilists
7. asymptotic time complexity
8. indexed sequential search
9. implementing linked lists using array
10. quadratic probing
11. circular linked list
12. discrete event simulation

二、简答题（每题 6 分，共 36 分）

1. 从概念上讲，树、森林和二叉树是三种不同的数据结构，将树、森林转化为二叉树的基本目的是什么，并指出树和二叉树的主要区别。
2. 面向对象的程序设计方法有何特点，并说明继承的含义。
3. 一棵二叉树的中序遍历序列为：ECBHFDJIGA，后序遍历序列为 ECHFJIGDBA，请构造出这棵二叉树，并写出它的先序遍历序列。
4. 线性表的顺序存储结构具有三个弱点：第一，在作插入或删除操作时，需要移动大量元素；第二，由于难以估计，必须预先分配较大的空间，往往使存储空间不能得到充分利用；第三，表的容量难以扩充。试问，线性表的链式存储结构是否一定能够克服上述三个弱点？请简述之。

5. 简要叙述 Hash 表技术中冲突的概念，并给出三种解决冲突的方法。

6. 何谓算法，其基本特性是什么？

三、填空（40 分）

1. 稀疏矩阵一般的压缩存储方法有两种，即_____ 和 _____。

2. 递归函数 f(n)=f(n-1)+n(n>1)的递归出口是_____，递归体是_____。将递归算法转换成对应的非递归算法时，通常需要使用_____。

3. 广义表(a， b，(c，d) ，e，((i，j)，k))的长度是_____，深度是_____；广义表运算式 HEAD(TAIL((x，y，z)，(a，b，c)))的结果为_____。

4. 以数据集(4，5，6，7，10，12，18)为结点权值所构造的哈夫曼树为_____，其带权路径长度为_____。

5. 已知序列{18，19，62，45，9，37，78，69，88}，采用快速排序法对该序列作升序排列时每一趟的结果为：

   初　始：_____

   第一趟：_____

   第二趟：_____

   第三趟：_____

   第四趟：_____

   第五趟：_____

   第六趟：_____

   第七趟：_____

   第八趟：_____

6. 对于长度为 n 的线性表，顺序查找法的平均查找长度为_____，其时间复杂度为____；二分查找法的平均查找长度为_____，其时间复杂度为_____；若采用分块查找（假定总块数和每块长度均接近$\sqrt{n}$），其时间复杂度为_____。

7. 在插入排序、希尔排序、选择排序、快速排序、堆排序、归并排序和基数排序中，排序是不稳定的有_____、_____、_____、_____，平均比较次数最少的排序是_____，需要内存容量最多的排序是_____。

8. 下面算法是实现对以邻接表存储的图进行深度优先遍历递归算法，请空白处填上适当的内容。

```
typedef enum {FALSE, TRUE} Boolean;
Boolean visited[MaxVertexNum];

void DFSTraverse(ALGraph *G)
{
    int i;
```

```
        for (i=0; i<G - >n; i++)
            visited[i] = _____;
        for (i=0; i<G - >n; i++)
            if (! visited[i])
                _____;
}
void DFS(ALGraph *G, int i)
{
    EdgeNode *p;
    printf("visit vertex; %c", G - >adglist[i], vertex);
    visited[i] = TRUE;
    p = G - >adglist[i].firstedge;
    while (_____){
        if (! visited[p - >adjvex])
            DFS(G, p - >adjvex);
        p = _____;
    }
}
```

四、改错（12 分）

1. 以下给出在链栈中实现插入操作的算法 Push，其类型和算法说明如下（2 处错误）：

```
typedef struct stacknode
{
   DataType data
   struct stacknode *next
}StackNode;
typedef StackNode *ListStack;
Void Push(LinkStack ls, DataType x)
{//将元素 x 插入链栈头部
    p= (StackNode * ) malloc (sizeof (StackNode));
    p - >number = x ;
    p - >next = ls ;
    ls= p - >data;
}
```

2. 以环形顺序队列的存储方式实现队列的基本算法如下（3 处错误）：

```
#include <iostream.h>
#define MaxLen 20
typedef char elemtype;
typedef struct
{
    elemtype data[MaxLen];
    int front, rear;
} queue;                    //front 为队头指针，rear 为队尾指针。
void init(queue *sq) //初始化队列
{
    sq - >front =0;
    sq - >rear = 0;
}
int inqueue (queue *sq, elemtype x)     //入队列
{
    if   ((sq - >rear+1) %MaxLen==sq - >front)     //队列上溢出
        return 0;
    else
    {
        sq - >rear=(sq - >rear+1) %MaxLen;
        sq - >data[sq - >rear+1]=x;
        return 1;
    }
}
int outqueue (queue *sq, elemtype *x)       //出队列
{
    if   ((sq - >front) ==sq - >rear)     //队列下溢出
        return 0;
    else
    {
        sq - >front=(sq - >front+1) %MaxLen;
        *x=sq - >data[sq - >front+1];
        return 1;
    }
}
int empty (queue *sq)     //判断队列是否为空队
{
    if   (sq - >rear) ==sq - >front)
        return 1;
```

```
        else
            return 0;
}
int gethead (queue *sq, elemtype *x)     //取队头
{
        if   (sq - >rear) ==sq - >front)
            return 0;     //队列下溢出
        else
        {
            *x=sq - >data[(sq - >front) %MaxLen];
            return 1;
        }
}
```

五、请按要求编写程序（50 分，可任选 PASCAL 或 C 语言实现）

1. 已知函数 M(x)定义如下（18 分）：

$$M(x)=\begin{cases} x-10 & x>100 \\ M(M(x+11)) & x\leqslant 100 \end{cases}$$

(1) 编写一个非递归函数计算给定 x 的 M (x)值；

(2) 编写一个递归函数计算给定 x 的 M(x)值。

2. 设计一种算法用于判定一棵给定的二叉树是否为完全二叉树（16 分）。

3. 设有文件“PERSONEL.TXT”存放职工的数据，该文件是用写字板编辑成的，其内容包括：职工号、姓名、性别、年龄、职称、基本工资、津贴、奖金、扣款、实发工资等（假设没有重复的职工号）。编写实现如下功能的函数（16 分）：

(1) 在 PERSONEL.TXT 文件末尾追加职工记录，其中基本工资、津贴、奖金、扣款由用户输入，而实发工资由计算机自动计算，即实发工资=基本工资+津贴+奖金-扣款；

(2) 根据用户输入的职工号和对应的数据修改该职工的数据；

(3) 根据用户输入的职工号删除该职工的数据；

(4) 根据用户输入的工资数，显示实发工资数额大于该工资数的职工的所有信息，并送往屏幕，其显示格式为：

职工号　姓名　性别　年龄　职称　基本工资　津贴　奖金　扣款　实发工资