广东工业大学
2006年攻读硕士学位研究生入学考试试题

考试科目(代fil5)名称：数据结构
满分：150
使用专业：计算机系统结构、计算机软件与理论、计算机应用技术

(考生注意：答卷封面需填写自己的准考证编号。答完后连同本试题一并交回!)
答题注意：请在答题纸上按题目顺序答题，在试题上答题无效!

一．填空题(共24分)

1．如果数据结构中的元素之间的关系是一个对多个的，则归类为_____结构

2．下列程序段的时间复杂度为_____

```
for(s=0, i=n; i>0; i-=2)
for(j=i / 2; j<I; j++)
S+=a[i][j]
```

3．在长度为n的有序链表中，求前趋结点的操作是O(_____)的.

4．递归函数是一个_____的函数.

5．与定长顺序存储结构相比，串的堆分配存储结构的主要优点是_____.

6．常用的稀疏矩阵压缩存储结构分别是_____和_____.

7．若森林非空，则对其中序遍历可按下述规则进行
(1)中序遍历森林中_____
(2)访问第一棵树的_____
(3)中序遍历森林中_____

8．二叉树的_____遍历算法需要使用队列作为辅助数据结构.

9．在合有n个顶点的连通图中，任意两个不同顶点之同的一条简单路径的边的数目最多是_____

10．除根结点之外．m阶B-树中的非终端结点至少有_____棵子树.

11．在归并排序中,”归并”的意义是_____.

12．在磁盘上读写一块信息所番的三部分时间分别是_____

二．单项选择题(共24分)

13.抽象数据类型是指一个数学模型以及定义在模型上的一组　　.
[A]元素　　[B]操作　　[C]关系　　[D]类型

14．链表不具有的特点是　　，
[A] 可以随机访问任一结点　　[B] 所需空问与表的长度成正比　　’
[C] 不必事先估计存储空问　　[D] 插入和删除不需要移动元素

15．在链队列的出队操作中，需要修改尾指针的条件是队列
[A] 原来已满　　[B] 原来已空　　[c] 变为满　　[D] 变为空

16．模式串”ababc”的next函数值为
[A] 00110　　[B] 01012　　[C] 01123　　[D] 12123　　-

17-二维效组A采用行优先的存储方法．其中每个元素占1个存储单元．若A[1][1]的地址

为 260，A[3][3]的地址为 286，则 A[5][5]的地址为
[A] 310　[B]311　[C]312　[D] 313
18．若一棵完全二叉树的第 6 层只有 6 个结点．则该树的叶子结点个数是
[A]6　[B]16　[C] 19　[D]不确定的
19．在一个带权连通围中，权值最小的边必定包含在该图的
[A]深度优先生成捌　[B]任一关键路径
[c]广度优先生成树　[D]任一最小生成树　。
20．对于无向图的存储结构，邻接多重表比邻接表更易于进行
[A]对边的操作　[B]圈的深度优先遍历
[C]对顶点的掇作　[D]图的广度优先遍历
21．适合进行高效率查找的动态查找表的组织结构是
[A]有序表　[B]分块有序表　[c]哈夫曼树　[D]二只排序树
22．希尔排序的增量序列必须是　。　。
[A] 递增的　[B]递减的　[C]等比的　[D] 等差的．
23．对记录序列(31 4，298，508，123，486，145)依次按个位和十位进行两趟基数排序之后所得结果为:
[A]123，145，298，314，486，508　[B]508，314，毕 3，H5，486，298
[C]486，31 4，123，145，508，298　[D]298，123，508，486，145，314
24．在 VSAH 文件的控制区问中．记录的存储方式为
[A]有序顺序　[B]无序顺序　[C]有序链接　[D] 无序链接

三．解答题(共 45 分)
25．(11 分)给定二叉树 T 如图所示．
(1)画出先序线索二叉树：
(2)画出中序前驱线索二叉树：
(3)画出后序后继线索二叉树．

26．(10 分)已知无向带权围 G 定义如下：
G=(V，E)
V={ a，b，c，d，e，f}
E={(a，b，4)，(a，c，3)，(b，c，5)，(b，d，5)，(b，e，7)，
(C，d，5)，(c，f，7)，(d，e，7)，(d，f，6)，(e，f，3)}
(1)画出 G 的邻接表：
(2)画出按克鲁斯卡尔算法求 G 的最小生成树的过程．

27．(8 分)设哈希函数为 H(k)=k％11,用二次探测法处理冲突.请画出依次插入元素 29,15,48,47,23,41,73,37 后,该哈希表的状态,在各元素下面标出其冲突次数,并求出查找成功的平均查找长度.

28.(8 分)请指出两种不稳定的排序算法,并分别给出一个不稳定的排序实例.

29.(7 分)已知某文件经过置换选择排序之后,得到长度分别为 48,9,13,39,18,5,25 和 7 的八个初始归并段.请为其构造 3 路平衡归并最佳归并树,并求出读写外存的次数.

四.算法题(共 57 分)

30. (7 分)阅读下列算法. 并回答问题

(1)设队列 P=(1,3, 5,2, 4, 6). 请写出执行算法 f30(P)后的队列 P

(2)简述算法 f30 的功能.

```
void f30(Queue&Q){
ElemType e;
If(!Empty(O)){
DeQueue(Q, e);
f30(Q);
EnQueue (Q, e);
}
}
```

31. (7 分)已知广义表类型 GList 和算法 f31 描述如下:

```
typedef enum { ATOM, LIST} ElemTag;
typedef struct GLNode {
ElemTag tag;
union{
char atom,
struct {
GLNode *hp, *tp;
}  ptr;
}  un;
} *GListj
void f31(GList A, int i){
if(A)
if(A 一>tag==ATOM)  printf(" (%c, %d)\n", A 一>un. atom, i);
else{
f31(A 一>un.ptr.hp, i+1);
f31(A 一>un.ptr.tp, i);
}
}
```

阅读算法 f31, 并回答问题

(1)设广义表 G=((((), a)), (b, (c)), d), 请写出执行算法 f31(G, 0)的全部输出;

(2)简述算法 f31 的功能.

32.(7 分) 阅读算法 f32，并回答问题

（1）设二叉排序树 bt 如右图所示，请写出执行算法 F32（bt，‘K’）的全部输出；

（2）简述算法 f32 的功能。

```
Void f32(BiTree  t, char x){
if (t!=NULL){
f32(t->rchild, x);
if(t->data >= x){
printf(“%c”,t->data);
f32(t->lchild, x);
}
}
}
```

33. (7 分)二叉树的存储结构的类型定义如下：

```
typedef Struct BiTNode{
char data;
BiTNode*ichild, *rchiId;
}BiTNode, *BiTree ;
```

已知某二叉树有 n 个结点，其前序序列和中序序列分别存于数组 Pre 和 ino 中. 算法 f33 建立

该二叉树的二叉链表 bt. 请在空缺处填入合适内容，使其成为完整的算法.

```
void f33(BiTree&bt, int ps, char*pre, int is, char*ino, int n)
/*当前要建立的子树 bt 的元素总数为 n. */
/*元素在前序序列 pre 的起始位置为 ps, */
/*元素在中序序列 ino 的起始位置为 is*/
{
int i=0 ;
Status found=false;
if(n==0)_____①_____;
else{
bt=(BiTNode*)malloc(sizeof(BiTNode));
bt 一>data=pre[ps];
while(!found＆＆i<n)
if(ino[is+i]= =_____②_____)found=true;
else i++;
if  (!found)bt=NULL;
else  {
if(i= =0)bt-> lchiId= NULL;
else_____③_____;
if  (_____④_____)   bt 一>rchiId=NULL;
elsef33(bt->rchiId, ps+i+1, pre, is+i+1, ino, n-i-1);
}
```

```
}
```

34. （7 分）假设堆定义为满足如下性质的完全三叉树：

(1) 空树为堆；

（2） 根结点的值不小于所有子树根的值，且所有子树均为堆。

堆采用顺序存储结构，其类型 HeapType 定义如下：

```
typedef  struct  {
char    r[MAXSIZE+1];
int      length;
}   HeapType;
```

算法 f34 将 h[s..m]调整为堆。请在空缺处填入合适内容，使其成为完整的算法。

```
void   f34(HeapType  &h、int  s、  int  m)    {
char  rc=h.r[s];
int   j;
for   (j=3*s;   j<=m;   __________①__________)    {
if (j<m)    {
if   (h.r[j-1] < h.r[j])    {
if   (h.r[j]   < h.r[j + 1])  __________②__________;

else  if   (h.r[j-1] < h.r[j+1])   j++;
else   j--;
}
else   if   (__________③__________)    j --;
if (re < h.r[j]){
h.r[s]=h.r[j];
_____④_____;
}   else  break;
}
h.r[s]=rc;
}
```

35. (7 分)已知有向图的邻接表类型定义如下：

```
typedef struct ArcNode {
int    adjvex;    / /邻接点域
struct ArcNode  *nextArc;    / /弧表链指针域
}ArcNode;      / /弧结点类型
typedef struct  1
char    verrex; / /顶点域
ArcNode    *firstArc; / /弧裹头指针
} VertexNode;      / /顶点表结点类型
typedef struct{
VertexNode vertices[MaxNum];    / /   顶点表
int  n，e;      / /   图中当前的顶点数和弧数
```

l ALGraph;　　/ /邻接表类型

算法 f35 利用 DFS 搜索策略在有向图 G 中寻找并输出经过顶点 vi 的所有简单回路. 数组 cyclePath 保存当前所求简单回路。如果 cyclePath[il=j，表示在回路中 Vi 的下一个顶点是 vj；如果 cyclePath【i】=一 1. 表示 vi 不在回路中. 请在空缺处填入合适内容，使其成为完整的算法。

```
int cyclePath[MaxNum]
void OutCycle(ALGraph G, int i){        / /输出简单回路
int j;
printf(" The Cycle is: %c,", Gvertices[i]. vertex);
for(j=cyclePath[i];  j! =i; j=cyclePath[j])
printf("%c,", G. vertices [j]. verrex);
printf("%C\n", G. vertices [i]. vertex);
}
void DFSPath(ALGraph G, int j, int i)I
ArcNode *P;
for  (p=G. verrices[J]. firstArc; p; p=p一>nextArc){
cyclePath[j] =_____①_____;
if(P->adjvex==i)_____②_____;
if(cyclePath[p->adjvex]== -1)_____③_____;
l
cyclePath[j]=-1;
void f35(ALGraph G, int i){
int j;
for(J=O; j<G. n; j++)   cyclePath[j]=_____④_____;
DFSPath(G, i, i);
}
```

36.(15 分)有序顺序表和二叉排序树的类型分别定义如下:

```
typedef struct {
char data[MAXSIZE];
int  length;
} SqList;   //顺序表类型
typedef struct BiTNode {
char      data;
BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

编写一个时间复杂度为 O(n)的算法,根据给定的有序顺序表 L,构造一棵平衡二叉排序树 T(注: 不设平衡因子, n=L.length)。