

2005 年上海理工大学硕士研究生入学考试题

考试科目: 数据结构与算法语言 准考证号: _____ 得分: _____

一、填空题 (每个空格 2 分, 共 20 分)

1. 设二维数组 $a[0 \dots m-1][0 \dots n-1]$ 按列优先顺序存储在首地址为 $\text{loc}(a_{00})$ 的存储区域中, 每个元素占 d 个单元, 则 a_{ij} 的地址为 _____。
2. 如果无向图 G 有 n 个顶点、 e 条边且用邻接矩阵进行存储, 那么深度优先遍历图 G 的时间复杂度为 _____。
3. 如果无向图 G 有 n 个顶点, 那么 G 的一棵生成树有且仅有 _____ 条边。
4. 对于一个具有 n 个元素序列如果采用快速排序, 那么所需最少比较次数的时间复杂度是 _____; 所需最大比较次数的时间复杂度是 _____。
5. 广义表 $((a,b),(c))$ 的表头是 _____, 表尾是 _____。
6. 对于一棵具有 n 个结点的二叉树, 若用二叉链表存储, 则空指针有 _____ 个。
7. 若一棵只有度为 0 和 2 的二叉树共有 51 个结点, 则二叉树的叶结点有 _____ 个。
8. 循环队列用数组 $A[0, m-1]$ 存放其元素, 已知其头尾指针分别是 front 和 rear , 则当前队列中的元素个数有 _____。

二、简答题 (每题 10 分, 共 30 分)

1. 假设通信电文使用的字符集为 $\{a,b,c,d,e,f,g\}$, 字符的哈夫曼编码依次为: 0110, 10, 110, 111, 00, 0111 和 010。画出此哈夫曼树, 并在叶子结点中标注相应字符。
2. 已知树 T 的先序遍历序列为 ABCDEFGHIJKL, 后序遍历序列为 CBEFDJIKLHGA。请画出树 T 。

3. 假定无向图 G 有 6 个结点和 9 条边,并依次输入这 9 条边 (1,2),(1,3),(1,5),(1,6),(2,3),(3,4),(3,5),(4,5),(5,6).请画出图 G 的邻接表.并按你所给的邻接表,写出从结点 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。

三、读程写结果（每题 10 分，共 20 分）

1. void fun (int n)
 {if (n>1&& n%2==1) fun (n-1);
 printf ("%3d",n);
 if (n>1&& n%2==0) fun (n-1);
 }
 写出调用语句 fun (5)的结果。

2. void f (int r[],int n)
 {int i,j,lastexchange,temp;
 i=n-1;
 while (i>0)
 {lastexchange=0;
 for(j=0;j<i;j++)
 if(r[j+1]<r[j])
 {temp=r[j]; r[j]=r[j+1];r[j+1]=temp; lastexchange=j;}
 i=lastexchange;
 }
 }

若 int A[6]={2, 7, 6, 3, 8, 9};写出调用语句 f(A,6)的结果。

四、设计题（每题 15 分，共 60 分）

1. 在文件 infile.txt 中按图 1 所示存放了若干个三角形的三边的长,编写程序: 逐个判断它们是否合理,并将合理的三角形面积计算出来,按图 2 所示存入文件 outfile.txt

3	4	5
6	8	10
...

图 1:infile.txt

edge1	edge2	edge3	area
3	4	5	6
6	8	10	24
...

图 2:outfile.txt

2. 设多项式 $p(x)=a_0+a_1x^1+a_2x^2+\cdots+a_nx^n$, 已知存储系数的数组, 多项式的幂次 n 及 x , 编写如下函数:

```
double poly(double a[], int n, double x)
```

3. 编写函数, 创建具有 n 个结点的带头单循环链表。

设链表存储结构如下:

```
typedef struct node
{
    int data;
    struct node *next;
} LNode;
```

4. 编写程序, 找出整数数组 $a[0..n-1]$ 中出现次数最多的整数。若有不同的整数出现同样的最多次数, 则找出其中最大的整数。

五、读程填充 (每个空格 2 分, 共 20 分)

1. 已知邻接表的顶点表结点结构为

vertex	firstedge
--------	-----------

边表结点 EdgeNode 的结构为

adjvex	next
--------	------

下列算法计算有向图 G 中顶点 i 的入度数。请在空缺处填入合适的内容, 使其成为一个完整的算法。

```
int FindDegree (ALGraph *G, int i) //ALGraph 为图的邻接表类型
{
    int degree, j;
    EdgeNode *p;
    degree = (1);
    for (j=0; j<G->n; j++)
    {
        p = G->adjlist[j].firstedge;
        while ((2))
        {
            if ((3))
                {degree++; break;}
            (4);
        }
    }
    return degree;
}
```

2. 完成下面对带头单链表进行排序的算法。

```
typedef struct lnode
{ int data;
  struct lnode *link;
} lnode, *linklist ;
void sort(linklist head)
{linklist p, q, r, s;
 p=head;
 while (p->link)
 {q=p->link; r=p;
  while ( (5) )
  {if(q->link->data<r->link->data)
    (6);
    (7)
  }
  if (r!=p) {s=r->link; (8);
             s->link=p->link; (9);}
  (10);
 }
}
```