

西安交通大学 2008 年攻读硕士学位研究生入学考试试题

考试科目: 计算机软件基础 科目编号: 831

考试时间: 1月20日下午

(注: 所有答案必须写在专用答题纸上, 写在本试题纸上和其它草稿纸上一律无效)

一、填空题 (20 分)

不写解答过程, 将正确答案与题目编号写在答题纸上。

- 1) 仅允许在同一端进行插入删除的线性表称为 ((1))。
- 2) 在一个长度为 n 的顺序表中第 i 个元素 ($1 < i < n$) 之前插入元素时, 需向后移动元素个数是 ((2))。
- 3) 将长度为 n 的单链表接在长度为 m 的单链表之后的算法时间复杂度为 ((3))。
- 4) 设数组 $data[m]$ 作为循环队列 SQ 的存储空间, $front$ 为队头指针, $rear$ 为队尾指针, 则执行出队操作后其头指针 $front$ 的值为 ((4))。
- 5) 对于前序遍历与中序遍历结果相同的二叉树为 ((5)); 对于前序遍历与后序遍历结果相同的二叉树为 ((6))。
- 6) 在用于表示有向图的邻接矩阵中, 对第 i 行的元素进行累加, 可得到第 i 个顶点的 ((7)) 度, 而对第 j 列的元素进行累加, 可得到第 j 个顶点的 ((8)) 度。
- 7) 已知一棵完全二叉树中共有 768 个结点, 则该树中共有 ((9)) 个叶子结点。
- 8) 堆排序是一种高效的内部排序算法, 它的时间复杂度是 ((10))。

二、判断题 (10 分)

- 1) 线性表的逻辑顺序与物理顺序总是一致的。
- 2) 线性表若采用链式存储表示时所有结点之间的存储单元地址可连续可不连续。
- 3) 中序遍历一棵二叉排序树的节点就可得到排好序的节点序列。
- 4) 直接选择排序是一种稳定的排序方法。
- 5) 负载因子 (装载因子) 是散列表的一个重要参数, 它反映散列表的装满程度。

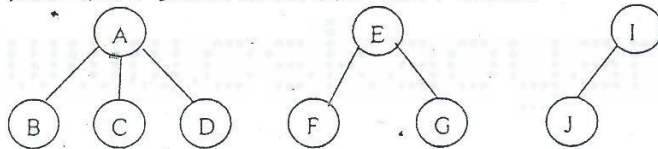
三、简答题 (16 分)

- 1) 简述顺序表与链表的优缺点。
- 2) 判断一个算法的优劣, 主要有哪几个标准? 简述其内容。
- 3) 什么是平衡二叉树?
- 4) 若有 n 个元素以构成一个小根堆 (也称小顶堆), 那么如果增加一个元素为 $K(n+1)$, 请用文字简要说明你如何在 $\log_2(n)$ 的时间内将其重新调整为一个堆?

四、解答题 (58 分)

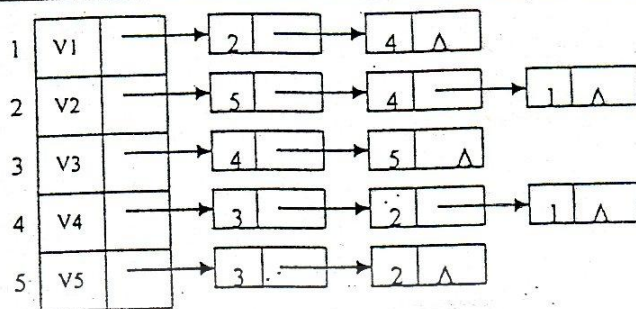
- 1) 循环队列 $squeue[45]$ 的头尾指针为 f, r , 请问该队列满的条件是什么? 当 $f=38, r=14$ 时, 该队列当前存储有多少元素? (6 分)

- 2) 如图一所示, 要求对该图的做如下工作: (10 分)



图一 森林

- (1) 将图一的森林转换为一棵二叉树;
- (2) 写出该二叉树的中序遍历和后序遍历的顺序。
- 3) 已知一个无向图的邻接表如图二所示。 (12 分)



图二 无向图的邻接表

- (1) 画出这个图。
- (2) 以 v_1 为出发点，对图进行广度优先搜索，写出所有可能的访问序列。
- (3) 以 v_1 为出发点，对图进行深度优先搜索，写出所有可能的访问序列。
- 4) 设 n 个元素的有序表为 R ， K 为一个给定的值，二分查找算法如下：

```
int binsearch( sqliist R, keytype K )
{
    j=1; h=n; suc=0;
    while((j<=h)&&(!suc))
    {
        mid=(j+h)/2;
        switch
        {
            case K=R[mid].key: suc=1; break;
            case K<R[mid].key: h=mid-1; break;
            case K>R[mid].key: j=mid+1;
        }
    }
    if(suc) return(mid); else return(0);
}
```

将上述算法中划线语句改为： $K < R[mid].key: h=mid$ 。

- (1) 改动后，算法能否正常工作？请说明原因。
 - (2) 若算法不能正常工作，给出一个查找序列和一个出错情况的查找键值；若能正常工作，请给出一个查找序列和查找某个键值的比较次数。(10分)
- 5) 从空树起，依次插入关键字 40, 8, 90, 15, 62, 95, 12, 23, 56, 32，构造一棵二叉排序树。(10分)

(1) 画出该二叉排序树

(2) 画出删去该树中元素值为 90 的结点之后的二叉排序树。

- 6) 有一组键值 25, 84, 21, 47, 15, 27, 68, 35, 24，采用快速排序方法由小到大进行排序，请写出每趟的结果，并标明在第一趟排序过程中键值的移动情况。(10分)

五、算法阅读题 (6分)

利用同一循环向量空间实现两个队列，其类型 Queue2 定义如下：

```
typedef struct {
    DataType data[MaxSize];
    int front[2], length[2];
} Queue2;
```

对于 $i=0$ 或 1 , $front[i]$ 和 $length[i]$ 分别为第 i 个队列的头指针和长度域。请在空缺处填入合适的内容，实现第 i 个循环队列的入队操作。


```

int EnQueue(Queue2*Q,int i,DataType x)
{//若第 i 个队列不满, 则元素 x 入队列, 并返回 1, 否则返回 0
    if(i<0||i>1)return 0;
    if(____(1)____)
        return 0;
    Q->data[____(2)____]=x;
    Q->length[____(3)____]++;
    return 1;
}

```

六、阅读下列函数说明和 C 程序, 将应填入____(n)____处的内容写在答题纸上。(10 分)

[说明] 本程序中的函数 factor(m, fac, cp) 用来计算正整数 m(m>2) 的除自身外所有不同因子的和。该函数返回因子和, 并把各不同因子按从小到大依次存放在数组 fac 中, 其因子个数存在形参指针变量 cp 所指的变量中。

[程序]

```

#include <stdio. h>
#define N 100

long factor(int m, ____ (1) ____,int *cp)
{
    int c1, c2, i, k;
    long s;
    fac[0] = 1;
    for(c1 = s = 1, c2 = N-1,i=2;; )
    { k = m/i;
        if(____ (2) ____ )
            if ( i != k)
            { fac[c1++] = i;
              fac[____ (3) ____] = k;
              s += i + k;
            }
            else {
                    fac[c1++] = i;
                    s += i;
                }
            i++;
            if (i >= k) ____ (4) ____;
        }
    for (c2++; c2<=N-1; c2++)
        fac[c1++] = fac[c2];
    *cp=c1;
    return(s);
}

main()
{
    int factors[N], i, count;
    long int sum;
    sum = factor( 16, factors, ____ (5) ____);
    for(i=0; i<count; i++)
        printf("factor is :%5d",factors[i]);
}

```

```

printf("\n\n");
printf("sum= %5ld    count= %5d\n",sum,count);
}

```

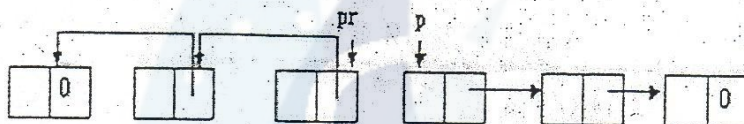
七、编程题 (30 分)

[说明] 以下各题只需写出实现算法的函数或过程即可, 不必写出整个程序, 可以使用 pascal 或 C 编写 (类 pascal 和类 C 均可), 也可以选用任意一种程序设计语言编写。必须写清楚算法设计思想及所用的数据结构; 对程序要加以适当的注解, 程序应有良好的结构, 不得使用 goto 语句。

1) 有 15 个人围成一圈, 顺序从 1 到 15 编号。从第一个人开始报数, 凡报到 n 的人退出圈子。用 C 语言写出程序, 输入 $n(n \geq 1)$ 的值, 输出最后留在圈子里的人的编号。

2) 假设有两个按元素值递增有序排列的线性表 A 和 B, 均以单链表作存储结构, 请编写算法将表 A 和表 B 归并成一个按元素非递减有序 (允许值相同) 排列的线性表 C, 并要求利用原表 (即表 A 和表 B) 的结点空间存放表 C。

3) 从左到右及从右到左遍历一个单链表是可能的, 其方法是在从左向右遍历的过程中将连接方向逆转, 如图三所示。在图中的指针 p 指向当前正在访问的节点, 指针 pr 指向指针 p 所指节点的左侧的节点。此时, 指针 p 所指节点左侧的所有节点的连接方向都已逆转。



图三

(1) 使用 Pascal 或 C 语言编写一个算法, 从任一给定位置 (pr, p) 开始, 将指针 p 右移 1 个节点。如果 p 移出链表, 则将 p 置为 NULL, 并让 pr 留在链表最右边的节点上。

(2) 使用 Pascal 或 C 语言编写一个算法, 从任一给定位置 (pr, p) 开始, 将指针 p 左移一个节点。如果 p 移出链表, 则将 p 置为 NULL, 并让 pr 停留在链表最左边的节点上。