复习题

一、单项选择题

1、 循环队列的队满条件为 （ ）

A. (sq.rear+1) % maxsize ==(sq.front+1) % maxsize;

B. (sq.front+1) % maxsize ==sq.rear

C. (sq.rear+1) % maxsize ==sq.front

D. sq.rear ==sq.front

2、 算术表达式 a+b*（c+d/e）转为后缀表达式后为（ ）

A. a a b * + a * c b * a / +　　B. a a * b + a * c b * a / +

C. a a b * a * c b * + a / +.　　D. a a b * + a c b * a / + *

3、 若对 n 阶对称矩阵 A 以行序为主序方式将其下三角形的元素(包括主对角线上所有元素)依次存放于一维数组 B [1..(n(n+1))/2] 中，则在 B 中确定 aij（i<j）的位置 k 的关系为（）。

A. i*(i-1)/2+j　　B. j*(j-1)/2+i　　C. i*(i+1)/2+j　　D. j*(j+1)/2+i

4、 分别用以下序列构造二叉排序树，与用其他三个序列构造的结果不同的是（ ）。

A.（ 100,80,90,60,120,110,130）　　B.（ 100, 120, 110,130,80, 60,90）

C.（ 100,60,80,90,120,110,130）　　D.（ 100,80, 60,90, 120, 130,110）

5、 设树 T 的度为 4，其中度为 1、2、3、4 的结点个数分别为 4、2、1、1。则 T 中有多少个叶子结点。

A．8　　B．10　　C．4　　D．6

6、 对关键码序列(28, 16, 32, 12, 60, 2, 5, 72)快速排序，从小到大一次划分结果为（ ）。

A．(2, 5, 12, 16) 28 (60, 32, 72)　　B．(5, 16, 2, 12) 28 (60, 32, 72)

C．(2, 16, 12, 5) 28 (60, 32, 72)　　D．(5, 16, 2, 12) 28 (32, 60, 72)

7、 设每个 d 叉树的结点有 d 个指针指向子树，有 n 个结点的 d 叉树有多少空链域？

A.nd　　B.n(d−1)　　C.n(d−1)+1　　D.以上都不是

8、 对于一非空的循环单链表，h 和 p 分别指向链表的头、尾结点，则有 （ ）。

A. p->next == h　　B. p->next == NULL

C. p == NULL　　D. p == h

9、 已知广义表 L=（(x,y,z)，a，(u，t，w)），从 L 表中取出原子项 t 的运算是（）。

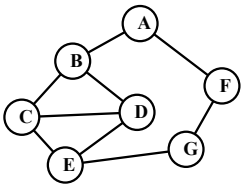
A. head（tail（tail（L）））　　B. tail（head（head（tail（L））））

C. head（tail（head（tail（L））））　　D. head（tail(head（tail（tail（L））））)

10、 已知串 S=“aaab”，其 next 数组值为（）

A. 0123　　B. 0112　　C. 0231　　D. 1211

11、 若如下图所示的无向连通图，则从顶点 A 开始对该图进行广度优先遍历，得到的顶点序列可能为(　　)。



A. A,B,C,D,E,F,G　　B. A,B,C,D,E,G,F

C. A,B,F,C,D,E,G　　D. A,B,F,C,D,G,E

12、 在用邻接表表示图时，拓扑排序算法时间复杂度为(　　)。

A. O(n)　　B. O(n＋e)　　C. O(n*n).　　D. O(n*e)

二、填空题

1. 循环队列用数组 A[0..m-1]存放其元素值，已知其头尾指针分别是 front 和 rear ，则当前队列的元素个数是__（rear-front+m）% m _____。 （ppt 循环队列求队列长度）

2. 设一棵完全二叉树具有 1000 个结点，则此完全二叉树有__500___个叶子结点，有 499 个度为 2 的结点

3. 先序次序访问序列为 ABCDEFGHIJKL，中序次序访问序列为 CBEFDGAJIKLH 的森林中有__2___棵树。(根据两个序列求出二叉树，再将二叉树转换成森林，得出森林中有 2 棵树)

4. 假定一个有向图的顶点集为{a, b, c, d, e, f}，边集为{<a, c>, <a, e>, <c, f>, <d, c>, <e, b>, <e, d>}，则出度为 0 的顶点个数为___2_____，入度为 1 的顶点个数为___2_____。

5. 若待排序的序列中存在多个记录具有相同的键值，经过排序，这些记录的相对次序仍然保持不变，则称这种排序方法是_稳定_____的，否则称为___非稳定____的。

6. 依次将关键字 5、6、9、13、8、2、12、15 插入初始为空的 4 阶 B 树后，根结点中包含的关键字是_6,9 _____。(ppt 上 b 树的插入操作，每次插入关键字都要插入在最后一层的节点上)

要构成连通的具有 n 个顶点的无向图，至少需要___n____条边，要连通具有 n 个顶点的有向图，至少需要___n-1____条边。（对于有向图来说，两个顶点之间的边是具有方向的，如果是构成连通的无向图，需要 n-1 条边，而对于有向图来说，只需要在第一个顶点和最后一个顶点加上一条边，让其构成环状的图即可，因此最少需要 n 条边。

7. 红黑树中，如果一个节点是红色的，则它的子节点必须是___黑色____的。

8. 执行折半查找法要求查找表必须为__顺序存储____结构。

9. 克鲁斯卡尔算法适合求___边稀疏____的网的最小生成树

三、应用题

1、 假设某段通信电文仅由 6 个字母 ABCDEF 组成，字母在电文中出现的频率分别为 2，3，7，15，4，6。根据这些频率作为权值构造哈夫曼编码，并给出最终构造出的哈夫曼树带权路径长度。(这里假定左节点的值小于右节点的值)

2、 将整数序列{40, 72, 38, 35, 27, 90, 75, 8, 55, 21}中的数依次插入到一棵空的平衡二叉树中，画出该树。

2、 假定一个待哈希存储的线性表为(32,75,29,63,48,95,25,46,14,70)，哈希地址空间为HT[13]，若采用除留余数法构造哈希函数 H(K)=K % 11 和线性探测法处理冲突，试画出该哈希表，并计算该各个元素的平均查找长度。

4、有一组数据（15，9，7，8，20，1，7，4），用堆排序的筛选方法建立初始小根堆。

五、算法填空

（1）统计二叉树度为 1 的结点个数。

```
int NodeCount ( BiTree T)
{
 if( _____(1)_____) return 0;
 if(T->lchild==NULL&&T->rchild!=NULL)
       return _____(2)_____ ;
 if(T->lchild!=NULL&&T->rchild==NULL)
       return _____(3)_____ ;
 _____(4)_____ ;
}
```

```
int NodeCount ( BiTree T)
{
 if( T==NULL 2 分 ) return 0;
 if(T->lchild==NULL&&T->rchild!=NULL)
  return NodeCount(T->rchild)+1 2 分 ;
 if(T->lchild!=NULL&&T->rchild==NULL)
  return NodeCount(T->lchild)+1 2 分 ;
  return NodeCount(T->lchild)+NodeCount(T->rchild) 2 分 ;
 }
```

（2）已知单链表 LA 和 LB 的元素按值非递减排列，归并 LA 和 LB 得到新的单链表 LC，LC 的元素也按值非递减排列。（链式有序表的合并）

```
void MergeList_L(LinkList &LA, LinkList &LB, LinkList &LC)
{
    LinkList pa, pb, pc;
    pa = LA->next;
    pb = LB->next;
    LC = LA;
    pc = LC;
    while( _____(1)_____)
    {
        if (pa->data <= pb->data)
        {
            _____(2)_____
            _____(3)_____
            _____(4)_____
        } else {
                _____(2)_____
                _____(3)_____
                _____(4)_____
        }
    }
    pc->next = pa ? pa : pb;
    delete LB;
}
```

```
void MergeList_L(LinkList &LA, LinkList &LB, LinkList &LC)
{
    LinkList pa, pb, pc;
    pa = LA->next;
    pb = LB->next;
    LC = LA;
    pc = LC;
    while( pa && pb 2 分 )
    {
        if (pa->data <= pb->data)
        {
            pc->next = pa;  2 分
            pc = pa;  2 分
            pa = pa->next;  2 分
        } else {
                pc->next = pb;  2 分
                pc = pb;  2 分
                pb = pb->next;  2 分
        }
    }
    pc->next = pa ? pa : pb;
    delete LB;
}
```