

2012~2013安徽大学《数据结构》期末试卷

一、单项选择题，在括号内填写所选择的标号（每小题1分，共12分）

1. 下面程序段的时间复杂度为()。

```
for(int i=0; i<m; i++)  
    for(int j=0; j<n; j++) a[i][j] = i*j;
```

A. $O(m^2)$ B. $O(n^2)$ C. $O(m*n)$ D. $O(m+n)$
2. 在二维数组中，每个数组元素同时处于()个向量中。
A. 0个 B. 1个 C. 2个 D. n个
3. 设有两个串t和p，求p在t中首次出现的位置的运算叫做()。
A. 求子串 B. 模式匹配 C. 串替换 D. 串连接
4. 利用双向链表作线性表的存储结构的优点是()。
A. 便于单向进行插入和删除的操作 B. 便于双向进行插入和删除的操作
C. 节省空间 D. 便于销毁结构释放空间
5. 设链式栈中结点的结构为(data, link)，且top是指向栈顶的指针。若想在链式栈的栈顶插入一个由指针s所指向的结点，则应执行()操作。
A. top->link=s; B. s->link=top->link; top->link=s;
C. s->link=top; top=s; D. s->link=top; top=top->link;
6. 设有一个递归算法如下

```
int X(int n) {  
    if(n<=3) return 1;  
    else return X(n-2)+X(n-4)+1;  
}
```

试问计算X(X(5))时需要调用()次X函数。
A. 2 B. 3 C. 4 D. 5
7. 一棵具有35个结点的完全二叉树的高度为()。假定空树的高度为-1。
A. 5 B. 6 C. 7 D. 8
8. 向具有n个结点的堆中插入一个新元素的时间复杂度为()。
A. $O(1)$ B. $O(n)$ C. $O(\log_2 n)$ D. $O(n \log_2 n)$
9. 在一棵AVL树中，每个结点的平衡因子的取值范围是()。
A. -1~1 B. -2~2 C. 1~2 D. 0~1
10. 一个有n个顶点和n条边的无向图一定是()。
A. 连通的 B. 不连通的 C. 无环的 D. 有环的
11. 在用Kruskal算法求解带权连通图的最小(代价)生成树时，通常采用一个()辅助结构，判断一条边的两个端点是否在同一个连通分量上。
A. 位向量 B. 堆 C. 并查集 D. 生成树顶点集合
12. 设有一个含有200个元素的表待散列存储，用线性探查法解决冲突，按关键码查询时找到一个元素的平均探查次数不能超过1.5，则散列表的长度应至少为()。
(注：平均探查次数的计算公式为 $S_{nl}=\{1+1/(1-\alpha)\}/2$ ，其中 α 为装填因子)
A. 400 B. 526 C. 624 D. 676

二、填空题，在横线处填写合适内容（每小题1分，共12分）

1. 数据结构的逻辑结构包括线性结构和_____结构两大类。
2. 在程序运行过程中不能扩充的数组是_____分配的数组。这种数组在声明它时必须指定它的大小。
3. 链表只适用于_____查找。
4. 设双向循环链表中每个结点的结构为(data, llink, rlink)，则结点*p的前驱结点的地址为_____。
5. 在一个链式队列中，若队头指针与队尾指针的值相同，则表示该队列至多有_____个结点。
6. 一棵树的广义表表示为a(b(c, d(e, f), g(h)), i(j, k(x, y))), 结点k的所有祖先的结点数为_____个。
7. 若将一棵树A(B(C, D, E), F(G(H), I))按照左子女-右兄弟表示法转换为二叉树，该二叉树中度为2的结点的个数为_____个。
8. 从一棵二叉搜索树中搜索一个元素时，若给定值大于根结点的值，则需要向_____继续搜索。
9. 设图G = (V, E)，其中

$$V = \{V_0, V_1, V_2, V_3\}$$

$$E = \{(V_0, V_1), (V_0, V_2), (V_0, V_3), (V_1, V_3)\}$$

则从顶点 V_0 开始对图G的深度优先序列总共有_____种。

10. 第*i* ($i=0, 1, \dots, n-2$) 趟从参加排序的序列中第*i*个至第*n-1*个元素中挑选出一个最小元素，把它交换到第*i*个位置，此种排序方法叫做_____排序。

11. 快速排序在最坏情况下的时间复杂度为_____。

12. 假定对长度*n*=100的线性表进行索引顺序搜索，并假定每个子表的长度均为 \sqrt{n} ，则进行索引顺序搜索的平均搜索长度为_____。

三、判断题，在每小题前面打对号表示正确或打叉号表示错误（每小题1分，共10分）

1. 算法和程序原则上没有区别，在讨论数据结构时二者是通用的。
2. 插入与删除操作是数据结构中最基本的两种操作，因此这两种操作在数组中也经常被使用。
3. 栈和队列都是顺序存取的线性表，但它们对存取位置的限制不同。
4. 将 $f=1+1/2+1/3+\dots+1/n$ 转化为递归函数时，递归部分为 $f(n)=f(n-1)+1/n$ ，递归结束条件为 $f(1)=1$ 。
5. 在一棵二叉树中，假定每个结点只有左子女，没有右子女，对它分别进行前序遍历和中根遍历，则具有相同的结果。
6. 进行折半搜索的表必须是顺序存储的有序表。
7. 用邻接矩阵存储一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小只与图中的顶点个数有关，而与图的边数无关。
8. 对于AOE网络，任一关键活动延迟将导致整个工程延迟完成。
9. 若将一批杂乱无章的数据按堆结构组织起来，则堆中数据必然按从小到大的顺序排列。
10. 一棵*m*阶B树中每个结点最多有*m-1*个关键码，最少有 $\lceil m/2 \rceil - 1$ 个关键码。

四、运算题（前2小题，每小题6分，后3小题，每小题8分，共36分）

1. 设有一个二维数组A[10][20]，按列为主序存放于一个连续的存储空间中，A[0][0]的存储地址是200，每个数组元素占1个存储字，则A[6][2]的存储字地址是多少。
2. 已知一棵二叉树的静态数组表示（即顺序存储）如下，其中-1表示空，请分别写出该二叉树的前序、中序、后序遍历序列。

0	1	2	3	4	5	6	7	8	9	10	11	12
20	8	46	5	15	30	-1	-1	-1	10	18	-1	35

3. 假定一组记录为(38, 42, 55, 15, 23, 44, 30, 74, 48, 26)，按次序插入每个记录生成一棵AVL树，给出最后得到的AVL树中度为2、度为1和度为0的结点个数。

4. 已知一个带权图的顶点集V和边集E分别为：

$$V = \{0, 1, 2, 3, 4, 5, 6\};$$

$$E = \{(0, 1)19, (0, 2)10, (0, 3)14, (1, 2)6, (1, 5)5, (2, 3)26, (2, 4)15, (3, 4)18, (4, 5)6, (4, 6)6, (5, 6)12\};$$

试根据迪克斯特拉(Dijkstra)算法求出从顶点0到其余各顶点的最短路径，即给出所经过的所有顶点。如顶点0到达顶点j需依次经过顶点k1和k2，则最短路径表示为0, k1, k2, j。

5. 已知有一个四元素的数据表{75, 75*, 60, 18}已经为最大堆，给出在堆排序过程中进行每一趟交换和调整后的数据表变化。

五、算法分析题（每小题6分，共18分）

1. 针对如下算法，回答问题：

若数组A[n]={12, 24, 0, 38, 0, 0, 0, 29, 0, 45, 0}，n=12，给出算法执行后数组A[n]的状态。

```
template <class T>
void unknown(T A[], int n) {
    int free=0;
    for(int i=0; i<n; i++)
        if(A[i]!=0) {
            if(i!=free) {
                A[free]=A[i];
```

```

        A[i]=0;
    }
    free++;
}
}

```

2. 针对如下算法，回答问题：

(1) 若整型数组A[8]={12, 24, 33, 38, 95, 83, 64, 57}， n=8，则给出算法返回的结果。

(2) 说明算法的功能是什么。

```

int unknown(int A[], int n) {
    if(n==1) return A[0];
    int temp=unknown(A, n-1);
    return A[n-1]>temp? A[n-1] : temp;
}

```

3. 已知二叉树中的结点类型BinTreeNode定义为：

```
struct BinTreeNode {ElemType data; BinTreeNode *left, *right;};
```

其中data为结点值域，left和right分别为指向左、右子女结点的指针域。参数bt指向一棵二叉树，引用参数x一开始具有的值小于树中所有结点的值。试根据下面的函数定义指出此算法的功能。

```

int JB(BinTreeNode* bt, ElemtType& x)
{
    if(bt==NULL) return 1;
    else {
        if(JB(bt->left, x)==0) return 0;
        if(bt->data<x) return 0;
        x=bt->data;
        if(JB(bt->right, x)==0) return 0;
        else return 1;
    }
}

```

六、算法设计题（每小题6分，共12分）

1. 设有两个整数类型的顺序表A（有m个元素）和B（有n个元素），其元素均以升序排列。把下面函数补充完整，将这两个顺序表合并成一个顺序表C，要求C的元素也以升序排列（表中允许元素重复）。

函数中的参数表给出参加运算的三个顺序表A、B与C。从C中得到执行结果。函数中用到顺序表的4个公有函数：

Length() 求表的当前长度；
maxLength() 求表的最大允许长度；
getData(int k) 提取第k个元素的值；
setData(int k, int val) 修改第k个元素的值为val。

```

template<class T>
void merge(SeqList<int>& A, SeqList<int>& B, SeqList<int>& C)
{
    int m=A.Length(), n=B.Length(), mpn=m+n;
    if(mpn>C.maxLength()) {
        cerr<< “合并后表的长度超出表C的最大允许长度” << endl;
        exit(1);
    }
    int i=0, j=0, k=0;
    int av=A.getData(i), bv=B.getData(j);
    //向下补充剩余的代码
}

```

2. 已知二叉树中的结点类型BinTreeNode定义为：

```
struct BinTreeNode {char data; BinTreeNode *left, *right;};
```

其中data为结点值域，left和right分别为指向左、右子女结点的指针域，根据下面函

数声明编写出删除一棵二叉树中所有结点的算法，并使树根指针为空。假定引用参数BT初始指向这棵二叉树的根结点。

```
void ClearBTree(BinTreeNode*& BT);
```