

试卷代号:1010

座位号

中央广播电视大学 2004—2005 学年度第二学期“开放本科”期末考试

计算机专业 数据结构 试题

2005 年 7 月

题号	一	二	三	四	五	六	总分
分数							

得分	评卷人

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. 执行下面程序段时,S 语句的执行次数为()。

```
for(int i=1;i<=n;i++)
    for(int j=1;j<=i;j++)S;
```

- A. n^2
- B. $n^2/2$
- C. $n(n+1)$
- D. $n(n+1)/2$

2. 多维数组实际上是由嵌套的()实现的。

- A. 一维数组
- B. 多项式
- C. 三元组表
- D. 简单变量

3. 不带头结点的单链表 first 为空的判定条件是()。

- A. `first == NULL;`
- B. `first->link == NULL;`
- C. `first->link == first;`
- D. `first! = NULL;`

4. 若让元素 1,2,3 依次进栈,则出栈次序不可能出现()种情况。

- A. 3,2,1
- B. 2,1,3
- C. 3,1,2
- D. 1,3,2

5. 设链式栈中结点的结构为(data,link),且 top 是指向栈顶的指针。若想摘除链式栈的栈顶结点,并将被摘除结点的值保存到 x 中,则应执行()操作。

- A. $x=top->data; top=top->link;$ B. $top=top->link; x=top->data;$
C. $x=top; top=top->link;$ D. $x=top->data;$

6. 设有一个广义表 A(a),其表尾为()。

- A. a B. (())
C. () D. (a)

7. 在一棵具有 n 个结点的完全二叉树中,树枝结点的最大编号为()。假定树根结点的编号为 0。

- A. $(n-1)/2$ B. $n/2$
C. $n/2+1$ D. $n/2-1$

8. 若搜索每个元素的概率相等,则在长度为 n 的顺序表上搜索任一元素的平均搜索长度为()。

- A. n B. $n+1$
C. $(n-1)/2$ D. $(n+1)/2$

9. 向一棵 AVL 树插入元素时,可能引起对最小不平衡子树的调整过程,此调整分为()种旋转类型。

- A. 2 B. 3
C. 4 D. 5

10. 为了实现图的广度优先搜索遍历,BFS 算法使用的一个辅助数据结构是()。

- A. 栈 B. 队列
C. 二叉树 D. 树

11. 若待排序序列在排序前已基本递增有序,则采用()方法比较次数最少。

- A. 直接插入排序 B. 快速排序
C. 归并排序 D. 直接选择排序

12. 5 阶 B 树中,每个结点最多允许有()个关键码。

- A. 2 B. 3
C. 4 D. 5

得分	评卷人

二、填空题,在横线处填写合适内容(每空 1 分,共 16 分)

1. 数据结构包括_____、_____和数据的运算三个方面。
2. 一维数组所占用的空间是连续的。但数组元素不一定顺序存取,通常是按元素的_____存取的。
3. 将一个 n 阶对称矩阵的上三角部分或下三角部分压缩存放于一个一维数组中,则该一维数组需要至少具有_____个元素。
4. 链接表与顺序表、_____表、_____表等一样都是数据逻辑结构的存储表示。
5. 在一个链式队列中,若队头指针与队尾指针的值_____,则表示该队列可能为空,也可能只包含有 1 个结点。
6. 对于一棵具有 n 个结点的树,该树中所有结点的度数之和为_____。
7. 在一棵高度为 3 的理想平衡二叉树中,最少含有_____个结点,最多含有_____个结点。假定树根结点的高度为 0。
8. 假定对长度 $n=50$ 的有序表进行折半搜索,则对应的判定树中最下一层的结点数_____个。
9. 用邻接矩阵存储图,占用的存储空间与图中的_____数有关。
10. 第 $i(i=1,2,\dots,n-1)$ 趟从参加排序的序列中取出第 i 个元素,把它插入到由第 0 个至第 $i-1$ 个元素组成的有序表中适当的位置,此种排序方法叫做_____排序。
11. 快速排序在平均情况下的时间复杂度和空间复杂度分别为_____和_____。
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时间复杂度为_____。

得 分	评卷人

三、判断题,在每小题后面的括号内打对号表示正确或打叉号表示错误(每小题 1 分,共 12 分)

1. 数据的逻辑结构与数据元素本身的内容和形式无关。 ()
2. 使用三元组表示稀疏矩阵中的非零元素能节省存储空间。 ()
3. 在使用后缀表示实现计算器类时用到一个栈的实例,它的作用是暂存运算对象或中间计算结果。 ()
4. 一个广义表的表头总是一个广义表。 ()
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按层遍历,则具有相同的结果。 ()
6. 能够在链接存储的有序表上进行折半搜索,其时间复杂度与在顺序存储的有序表上相同。 ()
7. 邻接表只能用于有向图的存储,邻接矩阵对于有向图和无向图的存储都适用。 ()
8. 在 AOE 网络中,可能同时存在几条关键路径,把所有关键路径都需通过的有向边称为桥。如果加速这样的桥上的关键活动就能使整个工程提前完成。 ()
9. 当输入序列已经基本有序时,起泡排序需要比较关键码的次数,比快速排序还要少。 ()
10. 在索引顺序结构上实施分块搜索,在等概率情况下,其平均搜索长度不仅与子表个数有关,而且与每一个子表中的对象个数有关。 ()
11. 若有 m 个初始归并段参加 k 路平衡归并排序,则归并趟数应为 $\lceil \log_2 m \rceil$ 。 ()
12. 向一棵 B 树插入关键码的过程中,若最终引起树根结点的分裂,则新树比原树的高度减少 1。 ()

相应长度:

5. 已知有一个四元素的数据表{30,18,20,15}已经为最大堆,给出堆排序过程中进行每一趟交换和调整后的数据表变化。

(0) {30 18 20 15}

(1)

(2)

(3)

得分	评卷人

五、算法分析题(每小题 6 分,共 18 分)

1. 设顺序表 SeqList 具有下列操作:

```
int Length( ) const;    //计算表长度并返回,若表为空则返回 0
T Remove( );           //删除当前表项并返回其值,置下一表项为当前表项
T First( );            //取表中第一个表项的值并返回,并置为当前表项
T Next( );              //取当前表项后继表项的值并返回,
                        //并把此后继表项置为当前表项
```

若顺序表中存放的数据为{29,38,47,16,95,64,73,83,51,10,0,26},表的长度为 12,参数值 $s=10, t=30$,说明算法执行后顺序表的状态和长度的变化。

```
#include<iostream. h>
#include"SeqList. h"
template<class T>
void unknown(SeqList<T>& L, T s, T t)
{
    if(! L. Length( )||s>=t)
        {cerr<<"表为空或参数值有误!"<<endl; exit(1);}
    int i=0;
    T temp=L. First( );
    while (i<L. Length( ))
        if (temp>=s && temp<=t)L. Remove( );
        else {temp=L. Next( );i++;}
}
```

算法执行后顺序表中的数据：

算法执行后顺序表的长度：

2. 针对如下算法, 设整数链表 L 中各结点的数据为 {12, 24, 30, 90, 84, 36}, n 的初值为 0,

则

(1) 给出执行 unknown(L.first, n) 调用后的返回结果；

(2) 指出算法功能。

在算法中 getLink() 函数返回结点指针域的值, getData() 函数返回结点的数据域的值。

```
float FF(ListNode<int> * f, int& n){
    if (f == NULL) return 0;
    else {
        n++;
        return FF(f->getLink(), n) + f->getData() / n;
    }
}
```

(1) 返回结果：

(2) 算法功能：

3. 已知一个有向图的顶点集 V 和边集 G 分别为：

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

$E = \{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \langle 3, 0 \rangle, \langle 4, 3 \rangle\};$

若它采用邻接矩阵存储, 对应 Graph 类的对象为 a, 则：

(1) 使用 a.count_D(3) 调用下面算法后的返回值是多少？

(2) 该算法的功能是什么？

(3) 给出该算法的时间复杂度。

```
Template<class Type>
int Graph<Type>::count_D(int i) {
    int d=0;
    for (int j=0; j<CurrentNode; j++) // CurrentNode 为图中的顶点数
    {
        if (Edge[i][j] != 0) d++;
        if (Edge[j][i] != 0) d++;
    }
}
```

```

    return(d);
}

```

(1)

(2)

(3)

得 分	评卷人

六、算法设计题(每小题 6 分,共 12 分)

1. 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求一棵二叉树中叶子结点总数的算法, 该总数值由函数返回。假定参数 BT 初始指向这棵二叉树的根结点。

```

int BTreeLeafCount(BinTreeNode * BT);

```

2. 假定下面函数 reArrange 通过扫描一遍 data 数组达到重新排列数据的目的, 使得所有负值数据位于所有非负值和 0 值数据之前。请补充完整 reArrange 函数体中遗漏部分, 即编写 while 循环体, 使其能够完成所要求的功能。(提示: 从两端向中间扫描)

```

template<class T>
void reArrange(T data[], int size){
    int i=0, j=size-1;
    T temp;
    while (i<j){ /* 此处遗漏了若干语句, 请将它们插入到下面 */
    }
    /* 在下面书写 while 循环体 */
}

```

试卷代号:1010

中央广播电视大学 2004—2005 学年度第二学期“开放本科”期末考试

计算机专业 数据结构 试题答案及评分标准

(供参考)

2005 年 7 月

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. D | 2. A | 3. A | 4. C | 5. A |
| 6. C | 7. D | 8. D | 9. C | 10. B |
| 11. A | 12. C | | | |

二、填空题,在横线处填写合适内容(每空 1 分,共 16 分)

1. 逻辑结构 存储结构
2. 下标(或顺序号)
3. $n(n+1)/2$
4. 索引 散列
5. 相同
6. $n-1$
7. 8 15
8. 19
9. 顶点
10. 直接插入
11. $O(n\log_2 n)$ $O(\log_2 n)$
12. $O(\sqrt{n})$

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 12 分)

- | | | | | |
|-------|-------|------|------|-------|
| 1. 对 | 2. 对 | 3. 对 | 4. 错 | 5. 对 |
| 6. 错 | 7. 错 | 8. 对 | 9. 对 | 10. 对 |
| 11. 错 | 12. 错 | | | |

四、运算题(每小题 6 分,共 30 分)

1. 41

答案说明:根据题意,矩阵 A 中当元素下标 I 与 J 满足 $I \geq J$ 时,任意元素 $A[I][J]$ 在一维数组 B 中的存放位置为 $I * (I+1)/2 + J$,因此, $A[8][5]$ 在数组 B 中位置为:

$$8 * (8+1)/2 + 5 = 41$$

2.

叶子结点数: 5 //2 分

单分支结点数:3 //2 分

两分支结点数:2 //1 分

三分支结点数:1 //1 分

3. 评分标准:每个结点的出度和入度值正确得 1 分,全对得 6 分。

结点	a	b	c	d	e
出度	1	1	2	1	2
入度	2	2	1	1	1

4. 评分标准:酌情给分

顶点: 2 3 1 5 4 6
最短长度: 10 14 16 21 25 31

5.

(0) {30 18 20 15}

(1) {20 18 15 30} //2 分

(2) {18 15 20 30} //2 分

(3) {15 18 20 30} //2 分

五、算法分析题(每小题 6 分,共 18 分)

1.

算法执行后顺序表中的数据:{38,47,95,64,73,83,51,0} //4 分

算法执行后顺序表的长度:8 //2 分

2.

(1)返回结果:46 //4分

(2)算法功能:求链表中各结点的值的平均值。 //2分

3.

(1)5 //2分

(2)返回顶点为 i 的度 //2分

(3)时间复杂度为 $O(n)$, n 为图中顶点数 //2分

六、算法设计题(每小题 6 分,共 12 分)

1. 分步给分

```
int BTreeLeafCount(BinTreeNode * BT)
{
    if (BT==NULL)return 0; //1分
    else if (BT->left==NULL && BT->right==NULL)return 1; //2分
    else return BTreeLeafCount(BT->left)+BTreeLeafCount(BT->right); //3分
}
```

说明:函数体中的两个 else 保留字可以省略

2. 分步给分

```
while (data[i]<0 && i<j) i++; //1分
```

```
while (data[j]>=0 && j>i) j--; //1分
```

```
if (i<j){
    temp=data[j];data[j]=data[i];data[i]=temp; //2分
```

```
}
i++;j--; //2分
```