

# 数据结构课程考试试题

考试时间： 2011 年 6 月 日 印刷份数： \_\_\_\_\_  
院系(部)： \_\_\_\_\_ 专业年级： 计算机及相关专业  
学 号： \_\_\_\_\_ 姓 名： \_\_\_\_\_

## 一、填空题(本大题共 15 小题 20 空, 每空 1 分, 共 20 分)

- \_\_\_\_\_是数据的基本单位, 在计算机程序中通常作为一个整体进行考虑和处理。
- 数据的逻辑结构是对数据之间关系的描述, 可分为集合、\_\_\_\_\_结构、树形结构和\_\_\_\_\_结构。
- 数据的存储结构是数据的逻辑结构在计算机存储器中的实现, 可分为顺序存储、\_\_\_\_\_存储、索引存储和\_\_\_\_\_存储 4 种方式。
- 抽象数据类型可表示为三元组 $(D, S, P)$ 表示, 其中  $D$  是数据对象,  $S$  是  $D$  上的关系,  $P$  是对  $D$  的\_\_\_\_\_。
- \_\_\_\_\_是对特定问题求解步骤的一种描述, 是指令的有限序列。
- 下面算法的时间复杂度是\_\_\_\_\_。

```
int Fac(int n)
{
    int f;
    if (n == 0) f = 1;
    else f = n * Fac(n - 1);
    return f;
}
```
- 只在表的一端进行插入和删除的线性表称为\_\_\_\_\_, 它的特点是\_\_\_\_\_。而在表的一端进行插入、另一端进行删除的线性表称为\_\_\_\_\_, 它的特点是\_\_\_\_\_。
- 在 C 语言中定义下面的二维实型数组:  
`double a[5][10];`  
每个元素占用 8 字节内存空间, 若数组起始地址为 0x1000, 则元素 `a[3][5]` 的地址为 0x\_\_\_\_\_。
- 将  $n$  阶的下三角矩阵采用压缩方式存储到一维数组中, 则元素  $a_{ij}$  ( $1 \leq i \leq n, 1 \leq j \leq n$ ) 对应一维数组元素的下标  $k$  ( $0 \leq k < n(n+1)/2$ ) 的计算公式为\_\_\_\_\_。
- 已知二叉树先根遍历的序列为“CDHAFEGB”, 中根遍历的序列为“HDFAECBG”, 则后根遍历的序列为“\_\_\_\_\_”。
- 已知完全二叉树有 1024 个结点, 则该二叉树的深度为\_\_\_\_\_。
- 具有 4 个结点且深度为 4 的二叉树有\_\_\_\_\_种形态。
- 依次将关键字 7、1、3、6、2、4、5 插入到空二叉排序树, 树的深度变为\_\_\_\_\_。
- 若无向图中度为 1 的顶点有 3 个, 度为 2 的顶点有 4 个, 度为 3 的顶点有 3 个, 度为 4 的顶点有 1 个, 则该无向图有\_\_\_\_\_条边。

15. 某无向图及其邻接表如图 1 所示，按深度优先搜索遍历所得到的结点序列为\_\_\_\_\_。

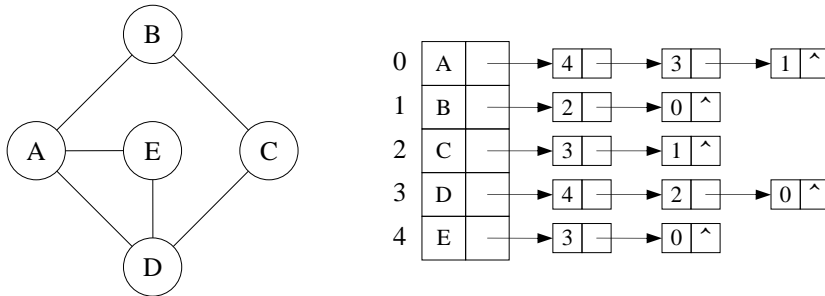


图 1 无向图及其邻接表

## 二、选择填空题(本大题共 15 小题，每小题 2 分，共 30 分)

- 算法的五个重要特性包括有穷性、确定性、\_\_\_\_\_、输入和输出。  
A) 可行性                      B) 健壮性                      C) 可读性                      D) 稳定性
- 算法分析的两个主要方面是\_\_\_\_\_。  
A) 空间复杂度和时间复杂度                      B) 正确性和简明性  
C) 可读性和文档性                      D) 数据复杂性和程序复杂性
- 下面是 4 种算法的时间复杂度，其中效率最高的是\_\_\_\_\_。  
A)  $O(2^n)$                       B)  $O(n^2)$                       C)  $O(\log_2 n)$                       D)  $O(n \log_2 n)$
- 在包含  $n$  个数据元素的顺序表中，假设每个位置  $i$  ( $1 \leq i \leq n$ ) 删除元素的机率相等，则在位置  $i$  处删除一个元素平均需要移动\_\_\_\_\_个元素。  
A) 1                      B)  $(n - 1) / 2$                       C)  $n / 2$                       D)  $n$
- 图 2 所示的双链表中，\_\_\_\_\_组语句将  $s$  所指结点插入到  $p$  所指结点的后面。

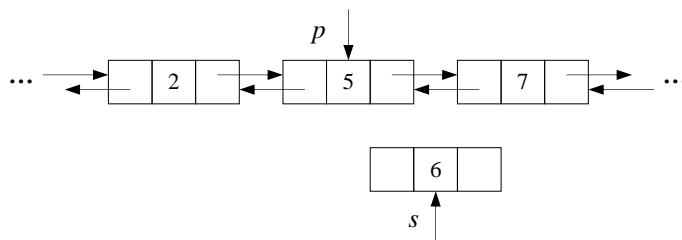


图 2 双链表

- |  |   |
|--|---|
| A) $p \rightarrow \text{next} \rightarrow \text{prior} = s;$ | B) $s \rightarrow \text{next} = p \rightarrow \text{next};$ |
| $s \rightarrow \text{prior} = p;$                            | $p \rightarrow \text{next} = s;$                            |
| $p \rightarrow \text{next} = s;$                             | $s \rightarrow \text{prior} = p;$                           |
| $s \rightarrow \text{next} = p \rightarrow \text{next};$     | $p \rightarrow \text{next} \rightarrow \text{prior} = s;$   |
| C) $s \rightarrow \text{next} = p \rightarrow \text{next};$  | D) $p \rightarrow \text{next} = s;$                         |
| $p \rightarrow \text{next} = s;$                             | $s \rightarrow \text{prior} = p;$                           |
| $s \rightarrow \text{next} \rightarrow \text{prior} = s;$    | $s \rightarrow \text{next} = p \rightarrow \text{next};$    |
| $s \rightarrow \text{prior} = p;$                            | $p \rightarrow \text{next} \rightarrow \text{prior} = s;$   |



三、画图题(本大题共 3 小题，每小题 5 分，共 15 分)

1. 将图 4 所示的二叉树按中根遍历的次序线索化，请画出对应的中根线索二叉树。

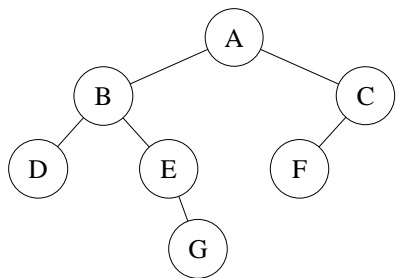


图 4 线索二叉树

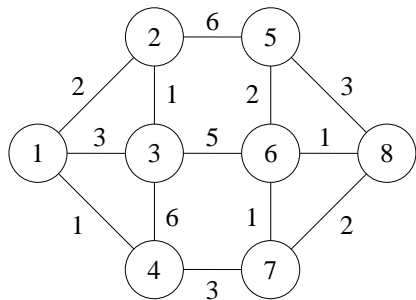


图 5 最小生成树

2. 依次将结点 4, 6, 7, 3, 2, 5, 1 插入到平衡二叉树中，请画出该平衡二叉树的动态插入和平衡的过程。
3. 按图 5 所示无向图，请画出 Prim 算法从顶点 1 出发求最小生成树的过程。

四、分析题(本大题共 4 小题，每小题 5 分，共 20 分)

1. 若线性表  $L = \{2, 3, 5, 6, 7, 0, 1, 8, 9, 4\}$ ，请写出用简单选择排序法按升序排序时，线性表变化过程的前 5 步。
2. 按图 6 所示有向图，请写出 Dijkstra 算法求从顶点 A 出发到其余顶点的最短路径的计算过程。

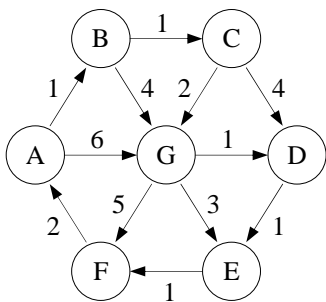


图 6 最短路径

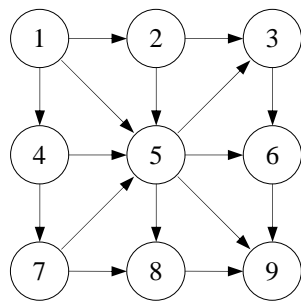


图 7 拓扑排序

3. 对图 7 所示的有向无环图进行拓扑排序，请写出至少 5 种排序结果。
4. 已知散列表表长为 15，地址计算公式为

$$H(k) = k \bmod 13$$

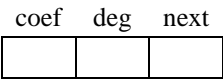
冲突处理方式为线性探测再散列，将关键字 5、31、18、15、2、3、16、4 依次插入到散列中，请写出散列表的状态。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

五、算法设计题(本大题共 2 小题，第 1 小题 9 分，第 2 小题 6 分，共 15 分)

1. 多项式求和(9 分)

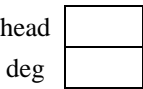
采用链式存储结构来表示多项式。其中结点的存储结构如下图所示：



结点的结构类型定义如下：

```
struct NODE
{
    double coef;           // 系数
    int deg;               // 次数
    struct NODE *next;     // 指针域
};
```

多项式的存储结构如下图所示：



多项式的结构类型定义如下：

```
struct POLY
{
    struct NODE *head;     // 头指针
    int deg;               // 多项式的次数(零多项式为-1)
};
```

多项式采用带头结点、按次数递减有序的单链表来表示。

设多项式:  $f(x) = 3x^4 - 2x + 1$ ,  $g(x) = x^5 + x^3 + 2x$ , 则它们的和为:

$$h(x) = f(x) + g(x) = x^5 + 3x^4 + x^3 + 1$$

如下图所示：

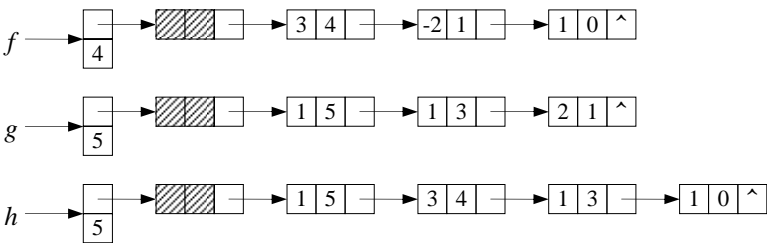


图 8 多项式求和

请编写算法，完成多项式的加法运算。函数原型如下：

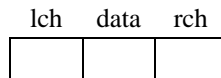
```
void AddPoly(struct POLY *h, const struct POLY *f, const struct POLY *g);
```

函数的参数都是指向多项式的指针，没有函数值。f 和 g 指向相加的两个多项式，h 指向准备保存结果的多项式。

要求：用文字描述算法思想，并估算时间复杂度，然后用 C/C++ 语言编码。

2. 求二叉树的深度(6 分)

二叉树采用链式存储结构，结点的存储结构如下图所示：



结点的结构类型定义如下：

```
struct NODE
{
    double data;           // 数据域
    struct NODE *lch, *rch; // 指针域
};
```

请用递归方法编写算法求二叉树的深度。

**int Depth(NODE \*root);**

函数的参数是根指针，函数值是二叉树的深度。如：

**printf("%d\n", Depth(root));**

要求：用文字描述算法思想，并估算时间复杂度，然后用 C/C++ 语言编码。