



# 厦门大学《数据结构》课程试卷

学院 \_\_\_\_\_ 系 2023 年级 \_\_\_\_\_ 专业 \_\_\_\_\_  
 学年学期: \_\_\_\_\_ 主考教师: \_\_\_\_\_ A 卷 (✓) B 卷 ( )

一、选择题 (每题 2 分, 共 20 分)

- (1) 如果数据元素之间存在一个对多个的关系, 则此结构为 ( )  
 A. 几何结构 B. 线性结构 C. 树形结构 D. 图状结构
- (2) 在下列序列中, 不是线性表的是 ( )  
 A. ("a", "b", "c") B. ("AB", "cd") C. ("a", true, "c") D. (a, b, c, d)
- (3) 在长度为  $n$  的顺序表中, 向第  $i$  个元素 ( $1 \leq i \leq n$ ) 之前插入一个元素时, 需向后移动 ( ) 个元素  
 A.  $n-i$  B.  $n-i+1$  C.  $n-i-1$  D.  $i$
- (4) 若某链表中常用的操作是在最后一个节点后插入一个节点和删除最后一个节点, 则 ( ) 存储方式最节省运行时间  
 A. 单链表 B. 带头节点的单链表 C. 单循环链表 D. 带头节点的双循环链表
- (5) 在解决计算机主机与打印机之间速度不匹配问题时通常设置一个打印数据缓冲区, 这样主机将要输出的数据依次写入缓冲区, 而打印机则从该缓冲区取出数据打印。该缓冲区是一个 ( ) 结构。  
 A. 栈 B. 队列 C. 数组 D. 线性表
- (6) 设按的最大长度为 3, 入栈序列为 1, 2, 3, 4, 5, 6, 则不可能的出栈序列是 ( )  
 A. 1, 2, 3, 4, 5, 6 B. 2, 1, 3, 4, 5, 6 C. 3, 4, 2, 1, 5, 6 D. 4, 3, 2, 1, 5, 6
- (7) ( ) 是 "abcd21ABCD" 的子串  
 A. abcd B. 321AB C. "abcABC" D. "21AB"
- (8) 一个 100 行, 100 列的三对角矩阵 (意思是除了第一行和最后一行外, 其余各行的非零元素是 3 个), 按行优先存入一维数组 B 中, 则 A 中元素 (第 66 行, 65 列) 在 B 中的位置是 ( ) (从 1 开始算)  
 A. 198 B. 195 C. 197 D. 200
- (9) 对二叉树的节点从 1 开始编号, 要求每个节点的编号大于其左右孩子的编号, 同一个节点的左右孩子中, 左孩子的编号小于右孩子的编号, 则可采用 ( )  
 A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
- (10) 3 个节点的不同形态的二叉树有 ( ) 种  
 A. 3 B. 4 C. 5 D. 6

二、填空题 (每空 2 分, 共 20 分)

- (1) 如果链表节点的指针域是 next, 则在非空线性链表中由 p 所指的节点后面插入一个由 q 所指向的节点的过程是执行 ① \_\_\_\_\_ ② \_\_\_\_\_
- (2) 一棵度为 2 的树与一棵二叉树有什么区别: \_\_\_\_\_
- (3) 已知度为  $k$  的树中有  $n_1$  个度为 1 的节点,  $n_2$  个度为 2 的节点, 以此类推,  $n_k$  个度为  $k$  的节点, 请问该树中有 \_\_\_\_\_ 个叶子节点
- (4) 用一维数组 A 存储顺序栈, 假设以 A[0] 为栈底, 以 top 作为栈顶指针, 出栈处理时,

(5) top 的变化操作为: \_\_\_\_\_

设目标为  $S="abcabcbcababababca"$ , 模式串为  $P="babab"$ , 写出空着的模式串 next 和 nextval 数值。next[4]=\_\_\_\_\_ nextval[5]=\_\_\_\_\_

	b	a	b	a	b
next	0	1	1	1	3
nextval	0	1	0	1	

- (6) 二维数组 A[cl:d1, c2:d2], 设每个元素占用 length 个存储单元, LOC(cl, c2) 是一个元素的存储位置, 则按行存放时, LOC(i, j)=\_\_\_\_\_
  - (7) 一个上三角矩阵 (即主对角线以下元素都是 0 的方阵) 按行优先存储, 则  $a_{ij} (i < j)$  的位置是 \_\_\_\_\_
  - (8) 一个三对角矩阵的元素个数为  $n$ , 按行压缩存储到一维数组中, 其长度为 \_\_\_\_\_
- 三、程序填空题 (每空 4 分, 共 20 分)
- (1) 递归求解汉诺塔
- ```

void Hanoi (int n, char x, char y, char z)
{
    if (n > 0)
        _____
        Move(x, n, z);
    }
    }
  
```
- (2) 如果 Getthead() 是取出广义表的表头, Gettail() 是取出广义表的表尾, 则从  $L=((a,b),(c,d))$  中, 取出 d 元素的表达式为 \_\_\_\_\_
  - (3) 皇后后的基本算法伪代码如下
- ```

void Trial (int i, int n) // n 行 n 列的棋盘前 i-1 行放置了符合约束的 i-1 个棋子, 从第 i 行开始
{
    if (i > n)
        输出棋盘的当前布置;
    Else
        {
            _____
            在第 i 行第 j 列放置一个棋子;
            移去第 i 行第 j 列的棋子;
        }
    }
  
```
- 四、画图题 (每题 5 分, 共 20 分)
- (1) 一个树的输入序列为 abcd, 请通过 Push 和 Pop 操作输出 cdba, 并画出过程。  
 提示: 时间不移, 可以把连续的 push 操作和连续的 pop 操作合并一张图, 并注明操作顺序
  - (2) 画出如下广义表的头尾链表存储结构: (((0, (a)), (b, c))). 其中表节点

1 hlink 指向表头 rlink 指向表尾

原子节点

0	原子节点的数据
---	---------

- (3) 给定先序遍历序列 ABDFCEG 和中序遍历序列 BFDAEGC, 画出二叉树
- (4) 给定一个权重集合为  $W=\{3, 15, 17, 14, 6, 16, 9, 2\}$ , 画出相应的赫夫曼树
- 五, 编程题 (每题 10 分, 共 20 分)
- (1) 使用一个非递归算法, 交换以二叉链表为存储结构的二叉树中所有节点的左右子树。  
要求: 使用下述数据结构 (可能需要自己定义其他的数据结构), 并写在 Exchange 函数体中。使用递归算法不得分。

```
typedef struct BiTNode {  
    TElemType    data;  
    struct BiTNode *lchild, *rchild; } BiTNode, *BiTree;  
void Exchange(BiTree *t){  
}
```

- (2) 试写出以孩子兄弟表示法的树计算森林的深度 (即森林中各棵树的深度最大值) 的算法。  
要求: 使用下述数据结构并写在 ForestDepth 函数体中。

```
typedef struct CSNode{  
    ElemType data;  
    struct CSNode *firstchild, *nextsibling;  
}CSNode, *CSTree;  
void ForestDepth (CSTree F, int &depth) {  
}
```