

2003-2004 学年第二学期

数据结构期终试卷 (A 卷)

(考试时间 100 分钟)

班级_____姓名_____学号_____得分_____

一、 单项选择题 (本大题共 15 小题, 每小题 2 分, 共 30 分) 在每小题列出的四个选项中只有一个符合题目要求, 请将其代码填在题后的括号内。错选或未选均无分。

1. 算法必须具备输入、输出和 [C]
A. 计算方法 B. 排序方法
C. 解决问题的有限运算步骤 D. 程序设计方法
2. 有 n 个节点的顺序表中, 算法的时间复杂度是 $O(1)$ 的操作是 [A]
A. 访问第 i 个节点 ($1 \leq i \leq n$)
B. 在第 i 个节点后插入一个新节点 ($1 \leq i \leq n$)
C. 删除第 i 个节点 ($1 \leq i \leq n$)
D. 将 n 个节点从小到大排序
3. 单链表的存储密度 [C]
A. 大于 1 B. 等于 1
C. 小于 1 D. 不能确定
4. 设将整数 1,2,3,4,5 依次进栈, 最后都出栈, 出栈可以在任何时刻 (只要栈不空) 进行, 则出栈序列不可能是 [B]
A. 23415 B. 54132
C. 23145 D. 15432
5. 循环队列 SQ 的存储空间是数组 $d[m]$, 队头、队尾指针分别是 $front$ 和 $rear$, 则执行出队后其头指针 $front$ 值是 [D]
A. $front=front+1$ B. $front=(front+1)\%(m-1)$
C. $front=(front-1)\%m$ D. $front=(front+1)\%m$
6. 在一个具有 n 个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是 [B]
A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n\log n)$
7. 设二维数组 $A[0..m-1][0..n-1]$ 按行优先顺序存储, 则元素 $A[i][j]$ 的地址为 [B]

- A. $LOC(A[0][0])+(i*m+j)$ B. $LOC(A[0][0])+(i*n+j)$
 C. $LOC(A[0][0])+[(i-1)*n+j-1]$ D. $LOC(A[0][0])+[(i-1)*m+j-1]$
8. 一个非空广义表的表头 [D]
 A. 一定是子表 B. 一定是原子
 C. 不能是子表 D. 可以是原子，也可以是子表
9. 具有 n 个节点的完全二叉树的深度为 [A]
 A. $\lceil \log_2(n+1) \rceil - 1$ B. $\log_2 n + 1$
 C. $\log_2 n$ D. $\lfloor \log_2 n \rfloor$
10. 若要惟一地确定一棵二叉树，只需知道该二叉树的 [D]
 A. 前序序列 B. 中序序列
 C. 前序和后序序列 D. 中序和后序序列
11. 在一个无向图中，所有顶点的度数之和等于图的边数的_____倍 [C]
 A. $1/2$ B. 1
 C. 2 D. 4
12. 拓扑排序运算只能用于 [C]
 A. 带权有向图 B. 连通无向图
 C. 有向无环图 D. 无向图
13. 在所有排序方法中，关键字比较的次数与记录的初始排列次序无关的是 [D]
 A. 希尔排序 B. 冒泡排序
 C. 插入排序 D. 选择排序
14. 下列排序算法中时间复杂度不受数据初始状态影响，恒为 $O(n^2)$ 的是 [C]
 A. 堆排序 B. 冒泡排序
 C. 直接选择排序 D. 快速排序
15. 二分查找要求节点 [A]
 A. 有序、顺序存储 B. 有序、链接存储
 C. 无序、顺序存储 D. 无序、链接存储

二、 填空题（本大题共 10 小题，每小题 2 分，共 20 分）不写解答过程，将正确的答案写在每小题的空格内。错填或不填均无分。

16. 数据的逻辑结构分为两大类，它们是线性结构和非线性结构。
17. 在单链表中（假设结点指针域名称为 next），删除指针 P 所指结点的后继结点的语句是 $p \rightarrow next = p \rightarrow next \rightarrow next$ 。
18. 已知循环队列用数组 data[n] 存储元素值，用 front, rear 分别作为头尾指针，则当前元素个数为 $(rear - front + n) \% n$ 。
19. 若 n 为主串长，m 为子串长，则串的朴素匹配算法最坏的情况下需要比较字符的总次数为 $(n - m + 1) \times m$ 。
20. 广义表((a),((b),j,(((d))))))的表尾是(((b),j,(((d))))))。
21. 已知二叉树有 61 个叶子节点，且仅有一个孩子的节点数为 45，则总节点数为166。
22. 解决计算机与打印机之间速度不匹配问题，须要设置一个数据缓冲区，应是一个队列结构。
23. n 个顶点 e 条边的图采用邻接表存储，深度优先遍历算法的时间复杂度为 $O(n + e)$ 。
24. 对于 n 个关键字的集合进行冒泡排序，在最坏情况下所需要的时间为 $O(n^2)$ 。
25. 在一个长度为 n 的顺序表中的第 i 个元素 ($1 \leq i \leq n$) 之前插入一个元素时，需向后移 $n - i + 1$ 个元素。

三、 解答题（本大题共 4 小题，共 25 分）

26. 对于下面的稀疏矩阵，画出其三元组法存储表示(假设下标从 0 开始)。(5 分)

$$\begin{bmatrix} 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 0 \\ 7 & 0 & 0 & 0 & 0 & 24 \\ 0 & 0 & 0 & 18 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 \end{bmatrix}$$

答案:

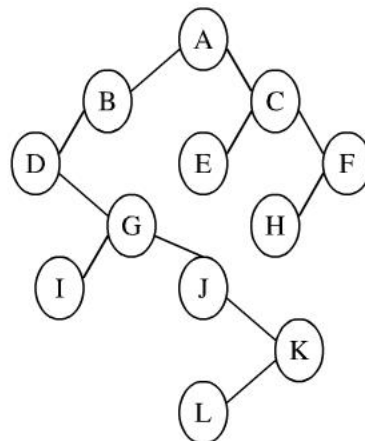
	行号	列号	值
0	0	2	14
1	1	4	-6
2	2	0	7
3	2	5	24
4	3	3	18
5	4	1	15

27. 已知一棵二叉树的中序序列和后序序列分别如下，请画出该二叉树。(5分)

中序序列: D I G J L K B A E C H F

后序序列: I L K J G D B E H F C A

答案:



28. 设有一个关键码的输入序列 {55,31,11,37,46,73,63,02,07}, (7分)

- (1) 从空树开始构造平衡二叉搜索树，画出每加入一个新结点时二叉树的形态。若发生不平衡，指明需做的平衡旋转的类型及平衡旋转的结果。(3分)
- (2) 计算该平衡二叉搜索树在等概率下的搜索成功的平均搜索长度和搜索不成功的平均搜索长度。(4分)

【解答】

(1) 构造平衡二叉搜索树的过程



(2) 计算在等概率下的搜索成功的平均搜索长度和搜索不成功的平均搜索长度。

$$ASL_{\text{succ}} = (1/9) * (1 + 2 * 2 + 3 * 4 + 4 * 2) = 25/9$$

$$ASL_{\text{unsucc}} = (1/10) * (3 * 6 + 4 * 4) = 17/5$$

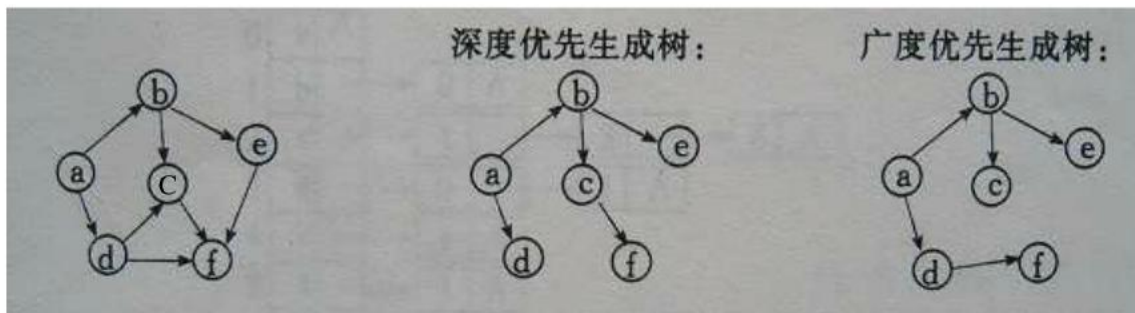
29. 已知一个图的邻接表如下所示。(8分)

(1) 画出该图的图形;(4分)

(2) 根据邻接表分别画出从顶点 a 出发进行深度优先和广度优先遍历所生成的生成树。(4分)

a	→	1	→	3	^
b	→	2	→	4	^
c	→	5	→	^	
d	→	2	→	5	^
e	→	5	→	^	
f	→	^	→		

答案:



四、 算法阅读题（本大题共 3 小题，每小题 5 分，共 15 分）

30. 设线性表的 n 个结点定义为 $(a_0, a_1, \dots, a_{n-1})$ ，在顺序表上实现的插入和删除算法如下，请在空白处填入适当内容。(顺序表的最大可容纳项数为 MaxSize)

```

Template <class Type> int SeqList<Type>::Insert(Type &x, int i) {
    If (i<0 || i>last+1 || last== ____ (1) ____ ) return 0;
    Else {
        Last++;
        For(int j=last;j>i;j--) data[j]=____ (2) ____;
        ____ (3) ____;
        Return 1;
    }
}

Template <class Type> int seqList<Type>::Remove(Type &x){
    int i=Find(x);
    if(i>=0) {
        last--;
        for (int j= ____ (4) ____;j<=last;j++) data[j]=____ (5) ____;
        return 1;
    }
    return 0;
}
    
```

答案:

- (1) $\text{MaxSize}-1$
- (2) $\text{data}[j-1]$
- (3) $\text{Data}[i]=x$
- (4) i
- (5) $\text{data}[j+1]$

31. 阅读下面的算法，请回答下列问题：

- (1) 试说明算法的功能。
- (2) 当执行该程序时，输入 12345678-1，输出什么结果？

```

#define StackSize 200
typedef int DataType;
typedef struct{
    DataType data[StackSize];
    int top;
}SeqStack;
void Push(SeqStack *s,DataType x)
{
    if(s->top!=StackSize-1)
        s->data[++s->top]=x;
}
DataType Pop(SeqStack *s)
{
    
```

```

        if(s->top!=-1)
            return s->data[s->top--];
    }
void main( )
{
    DataType i;
    SeqStack s;
    s.top=-1;
    scanf("%d",&i);
    while(i!=-1)
    {
        push(&s,i);
        scanf("%d",&i);
    }
    while(s->top!=-1)
    {
        i=Pop(&s);
        printf("%6d",i);
    }
}

```

答案:

(1)程序的功能是把输入的一串整数(用-1 做结束标记) ， 按照与输入相反的次序输出。
用栈实现这一功能。

(2) 输出结果 8 7 6 5 4 3 2 1。

32. 已知二叉树的存储结构为二叉链表，阅读下面算法。说明该算法的功能。

```

Template<class Type>
int BinaryTree<Type>::height(BinTreeNode<Type> *t){
    if(t==NULL) return -1;
    int h1=height(t->leftChild);
    int hr=height(t->rightChild);
    return 1+(h1>hr?h1:hr);
}

```

答案： 该算法的功能是统计二叉树的高度。

五、 算法设计题（本题共 10 分）

33. 设一棵二叉树以二叉链表表示，试以成员函数形式编写有关二叉树的递归算法。
- (1) 统计二叉树中度为 1 的结点个数。（5 分）
 - (2) 统计二叉树中度为 2 的结点个数。（5 分）
- （提示：递归算法如 32 题所示）

解答：(1) 统计二叉树中度为 1 的结点个数。

```

Template<class Type>
Int BinaryTree<Type> ::Degree1(BinTreeNode<Type> *t)const{
    If(t==NULL) return 0;
    If(t->leftchild!=NULL && t->rightchild==NULL || t->leftchild==NULL && t->rightchild!=NULL)
        Return 1+Degree1(t->leftchild)+Degree1(t->rightchild);
    Return Degree1(t->leftchild)+Degree1(t->rightchild);
}
    
```

(2) 统计二叉树中度为 2 的结点个数。

```

Template<class Type>
Int BinaryTree<Type> ::Degree2(BinTreeNode<Type> *t)const{
    If(t==NULL) return 0;
    If(t->leftchild!=NULL && t->rightchild!=NULL )
        Return 1+Degree2(t->leftchild)+Degree2(t->rightchild);
    Return Degree2(t->leftchild)+Degree2(t->rightchild);
}
    
```