

任课教师: 高宏宾、白明、金旺春、刘兴林、邢润丹 试卷分类(A 卷或 B 卷) A

## 五邑大学 试 卷 参 考 答 案 及 评 分 标 准

学期: 2016 至 2017 学年度 第 2 学期

课程: 数据结构 课程代号: 0800610 使用班级: 150801~08、10

一、

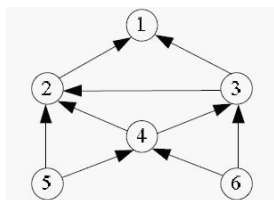
得分	
----	--

 单项选择题 (10 小题, 每小题 1 分, 共 10 分)

1. 下面关于线性表的叙述错误的是( **D** )。
  - A. 线性表采用顺序存储必须占用一片连续的存储空间
  - B. 线性表采用链式存储不必占用一片连续的存储空间
  - C. 线性表采用链式存储便于插入和删除操作的实现
  - D. 线性表采用顺序存储便于插入和删除操作的实现
2. 设输入序列为 1、2、3、4、5、6, 则通过栈的作用后可以得到的输出序列为( **B** )。
  - A. 5, 3, 4, 6, 1, 2
  - B. 3, 2, 5, 6, 4, 1
  - C. 3, 1, 2, 5, 4, 6
  - D. 1, 5, 4, 6, 2, 3
3. 二叉树的第  $k$  层的结点数最多为( **C** )。
  - A.  $2^k$
  - B.  $2^{k+1}$
  - C.  $2^{k-1}$
  - D.  $2(k-1)$
4. 在一个单链表中, 已知  $q$  所指结点是  $p$  所指结点的直接前驱, 若在  $q$  和  $p$  之间插入  $s$  所指结点, 则执行( **B** )操作。
  - A.  $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$
  - B.  $q \rightarrow \text{next} = s; s \rightarrow \text{next} = p;$
  - C.  $p \rightarrow \text{next} = s \rightarrow \text{next}; s \rightarrow \text{next} = p;$
  - D.  $p \rightarrow \text{next} = s; s \rightarrow \text{next} = q;$
5. 对于  $n$  个关键字序列  $\{k_1, k_2, \dots, k_n\}$ , 当且仅当满足关系  $k_i \leq k_{2i}$  且  $k_i \leq k_{2i+1}$  ( $2i \leq n, 2i+1 \leq n$ ) 称其为小根堆, 反之, 当且仅当满足关系  $k_i \geq k_{2i}$  且  $k_i \geq k_{2i+1}$  ( $2i \leq n, 2i+1 \leq n$ ) 则为大根堆, 以下序列中( **C** )不符合堆的定义。
  - A. (4,10,15,72,39,23,18)
  - B. (58,27,36,12,8,23,9)
  - C. (4,10,18,72,39,23,15)
  - D. (58,36,27,12,8,23,9)
6. 下面程序的时间复杂为( **B** )。

```
for (i=1, s=0; i<=n; i++)
{
    t=1;
    for(j=1; j<=i; j++)
        t=t*j;
    s=s+t;
}
```

  - A.  $O(n)$
  - B.  $O(n^2)$
  - C.  $O(n^3)$
  - D.  $O(1)$
7. 表达式  $a*(b+c)-d$  的后缀表达式是( **A** )。
  - A.  $abc+*d-$
  - B.  $abcd*+-$
  - C.  $abc*+d-$
  - D.  $-+*abcd$
8. 设无向图  $G$  中有  $n$  个顶点  $e$  条边, 则其对应的邻接表中的表头结点和表结点的个数分别为( **D** )。
  - A.  $n, e$
  - B.  $e, n$
  - C.  $2n, e$
  - D.  $n, 2e$
9. 拓扑序列是无环有向图中所有顶点的一个线性序列, 图中任意路径中的各个顶点在该图的拓扑序列中保持先后关系, ( **A** )为下图所示有向图的一个拓扑序列。



- A. 6 5 4 3 2 1  
B. 1 2 3 4 5 6  
C. 5 6 3 4 2 1  
D. 5 6 4 2 1 3

10. 数组  $A[0..5,0..6]$  的每个元素占 5 个字节，将其按列优先次序存储在起始地址为 1000 的内存单元中，则元素  $A[5, 5]$  的地址是( A )。

- A. 1175                      B. 1180                      C. 1205                      D. 1210

二、

得分	
----	--

 简单填空题（10 小题，20 个空，每空 1 分，共 20 分）

1. 数据的逻辑结构是( 数据元素及其相互关系 )。数据的物理结构是数据的逻辑结构在( 计算机内存中的表示形式 )。
2. 在长度为  $n$  的数组  $a[n]$  中（下标从 0 开始）存储两个栈时，假设左栈的栈顶指针（下标）是  $topl$ ，右栈的栈顶指针是  $topr$ ，那么两个栈均满的条件是(  $topl==topr-1$  )，右栈为空的条件是(  $topr==n$  )。
3. 假设将线性表  $L$  存储在以  $head$  为头指针的不带头结点的单链表中，则当(  $head==NULL$  )时，表示线性表  $L$  是空表。假设将线性表  $L$  存储在以  $head$  为头指针的带头结点的单链表中，则当(  $head->next==NULL$  )时，表示线性表  $L$  是空表。
4. 在长度为  $n$  的数组  $a[n]$  中（下标从 0 开始）存储一个循环队列时，假设队头指针（下标）是  $front$ ，队尾指针是  $rear$ ，而且  $a[(front+1)\%n]$  是队头元素， $a[rear]$  是队尾元素，那么队空的条件是(  $rear==front$  )，队满的条件是(  $(rear+1)\%n==front$  )。
5. 对于一个  $n*n$  的下三角数组  $a$ ，按照行优先的原则，把  $a$  的下三角部分的元素存储在起始地址为  $start$  的连续的内存空间中，假定数组的下标从 0 开始，一个数组元素占内存的  $c$  个字节，那么一共要存储(  $(1+n)*n/2$  )个数组元素，下三角部分的一个数组元素  $a[i][j]$  的地址是(  $start+((1+i)*i/2+j)*c$  )。
6.  $((a,b,c), d, (e, (x, y)), ( ))$  是一个长度为( 4 )，深度为( 3 )的广义表。
7. 含有  $n$  个结点的完全二叉树的深度是(  $\lfloor \log_2 n \rfloor + 1$  )。如果从 1 开始，按照层序顺序对完全二叉树的结点进行编号，则编号为  $i$  的结点有左孩子的话，它的左孩子的编号一定是(  $2*i$  )。
8. 在树的二叉树表示中，一个结点与它的左孩子是( 父子关系 )，与它的右孩子是( 兄弟关系 )。
9. AOV-网是一个( 顶点表示活动，有向边表示活动之间优先关系 )的有向图。如果一个 AOV-网的拓扑序列包含了它的所有顶点，表明这个 AOV-网中( 没有有向回路 )。
10. 二叉树中一个结点的平衡因子是( 它的左子树深度与它的右子树深度之差 )。如果一个二叉树是平衡二叉树，则( 它的所有结点的平衡因子的绝对值均小于等于 1 )。

三、

得分	
----	--

 应用题（10 小题，每小题 6 分，共 60 分）

1. 在如下数组 A 中存储了一个静态链表，表头指针为 A[0].next，试写出该线性表。

A	0	1	2	3	4	5	6	7
data		60	50	78	90	34		40
next	3	5	7	2	-1	4		1

**答案：线性表为：(78, 50, 40, 60, 34, 90)，顺序正确每个一分**

2. 一棵有 n 个结点的满二叉树有 ( ) 个度为 1 的结点，有 ( ) 个分支 (非终端) 结点和 ( ) 个叶子结点，该满二叉树的深度为 ( )。

**答案： 0, (n-1)/2 (n+1)/2  $\lfloor \log_2 n \rfloor + 1$**

**评分标准：1 分 1 分 1 分 3 分**

3. 设完全二叉树的顺序存储结构中存储数据 ABCDE，该二叉树的前序遍历序列为( )、中序遍历序列为( )和后序遍历序列为( )。

**答案：前序 ABDEC 中序 DBEAC 后序 DEBCA**

**评分标准：2 分 2 分 2 分**

4. 设给定一个权值集合  $W=(3, 5, 7, 9, 11)$ ，要求根据给定的权值集合构造一棵哈夫曼树，该哈夫曼树的带权路径长度  $WPL=( )$ ，元素 9 的哈夫曼编码为 ( )。

**答案：  $WPL=(3+5)*3+(7+9+11)*2=78$  10**

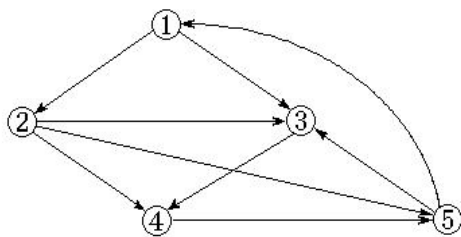
**评分标准：4 分 2 分**

5. 设一组初始记录关键字集合为(25, 10, 8, 27, 32, 68)，散列表的长度为 8，散列函数  $H(k)=k \% 7$ ，用线性探测解决冲突，所以元素查找成功的  $ASL=( )$ ，( )查找次数最多。

**答案： 9/6 68**

**评分标准：4 分 2 分**

6. 对于题 6 图所示的有向图若存储它采用邻接表，并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的，试写出：



题 6 图

(1) 从顶点①出发进行深度优先搜索所得到的深度优先生成树；

(2) 从顶点②出发进行广度优先搜索所得到的广度优先生成树；

**答案 评分标准：**

**(1) DFS: 1 2 3 4 5 3 分**

**(2) BFS: 2 3 4 5 1 3 分**

7. 2. 已知一个图的顶点集 V 和边集 E 分别为：

$V=\{1,2,3,4,5,6,7\};E=\{(1,2)3,(1,3)5,(1,4)8,(2,5)10,(2,3)6,(3,4)15,(3,5)12,(3,6)9,(4,6)4,(4,7)20,(5,6)18,(6,7)25\}$ ; 用克鲁斯卡尔算法得到最小生成树, 在生成最小生成树过程中依次得到的各条边为 ( )。

**答案: 用克鲁斯卡尔算法得到的最小生成树为:**

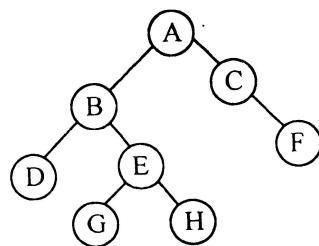
**(1,2)3, (4,6)4, (1,3)5, (1,4)8, (2,5)10, (4,7)20**

**评分标准: 各 1 分**

8. 设一组初始记录关键字序列为(15, 17, 18, 22, 35, 51, 60, 65), 按折半查找算法进行查找, 每个元素成功查找时的平均查找长度为( )。

**答案: 21/8**

9. 如图题 9 图所示的二叉树, 将其转换成对应的森林, 问: 该森林中共有( )棵树, ( )结点的度最大。

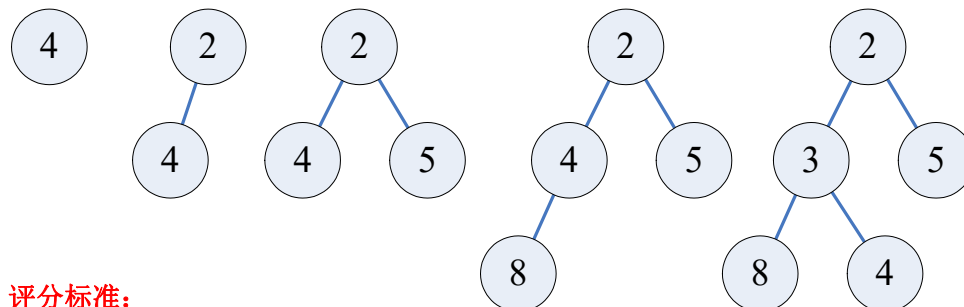


**答案: 3 A**

**评分标准: 各 3 分**

10. 在小根堆的操作中设有插入操作, 每插入一个元素都是放在尾部, 然后调整成小根堆。设初始小根堆为空, 画出小孩小根堆顺序插入数据 4, 2, 5, 8, 3 时, 每插入一个数据后堆的变化图。

**答案:**



**评分标准:**

**1 分**

**1 分**

**1 分**

**1 分**

**2 分**

四、

得分	
----	--

 算法设计题 (2 小题, 每小题 5 分, 共 10 分)

1. 一个顺序表, 存放在数组  $a[]$  中, 元素的类型为整型 (int), 顺序表的长度为  $n$ 。设计算法调整  $a$ , 使其左边的所有元素小于 0, 右边的所有元素大于或等于 0 (要求算法的时间复杂度为  $O(n)$ , 空间复杂度为  $O(1)$ )。

**评分: 算法的时间复杂度  $> O(n)$  扣 1 分, 空间复杂度  $> O(1)$  扣 1 分, 具体评分见程序标注, 多解请参照评分。**

`void quickSwapList(int a[],int n)`

```

{
    int i = 0, j = n - 1; // n 为顺序表的长度，即数的个数
    while( i < j ) //初值1分
    {
        while( a[i] < 0 && i < j ) i ++ ; // 找到左边大于等于 0 的数 //1分
        while( a[j] >= 0 && i < j ) j -- ; // 找到右边小于 0 的数 //1分
        if( i < j ) // 交换
        {
            int t = a[i] ; a[i] = a[j] ; a[j] = t ;
        } //以上1分
        i ++ ;
        j -- ; //循环控制1分
    }
}

```

2. 对于给定的简单算术表达式（只包含“+”、“-”、“\*”、“/”和“（）”运算），存放在字符数组 `expr[]` 中，表达式的长度为 `n`。从左到右逐个字符进行扫描，利用栈（已提供的栈和基本的栈操作，并假设不会出现栈满情况）设计一个判断圆括号（‘（’和‘）’）是否配对（每个右括号与最近遇到的尚未匹配的左括号相匹配）的算法，输出配对情况（配对/当前左比右少/左比右多）。

**评分：没有利用栈扣 2 分，具体评分见程序标注，多解请参照评分。**

栈：Stack \*s，基本操作如下：

栈的初始化：Stack \* StackInit( ); //构造一个空栈

入栈：void Push(Stack \*s ,char x); //将数据元素 x 插入栈顶

出栈：void Pop(Stack \*s ); //删除栈顶元素

判栈空：int StackEmpty(Stack \*s ); //栈空返回 TRUE（或 1）；否则返回 FALSE（或 0）

void Parenthesis(char expr[ ], int n)

```

{
    int i;
    Stack s=StackInit( ); //以上 1 分
    for(i=0; i<n; i++)
    {
        if(expr[i]=='(') Push(s ,expr[i]);
        else if(expr[i]==')')
        {
            if(!StackEmpty(s)) Pop(s );
            else { printf("当前左比右少\n" );return;}
        }
    } //以上 3 分
    if(!StackEmpty(s)) printf("当前左比右多\n" );
    else printf("匹配\n" ); //以上 1 分
}

```