

NO.3

一、 判断题（每小题 1 分，共 10 分）

1. 索引存储方法：除建立存储结点信息外，还建立附加的索引表来标识结点的地址。组成索引表的索引项由结点的关键字和地址组成。若每个结点在索引表中都有一个索引项，则该索引表称之为稠密索引（Dense Index）。若一组结点在索引表中只对应一个索引项，则该索引表称为稀疏索引。
2. 在双链表中，若仅知道指针 p 指向某结点，不知道头指针，能将结点 $*p$ 从相应的链表中删去，其时间复杂度为 $O(n)$ 。
3. 用 I 表示入栈操作， O 表示出栈操作，若元素入栈的顺序为 A, B, C, D, E ，为了得到 C, B, A, D, E 的出栈顺序，相应的 I 和 O 的操作序列为 $IIIOOOIOIO$ 。
4. 使用一个 40 个元素的数组存储循环队列，如果采取少用一个元素空间的方法来区别循环队列的队空和队满，约定队头指针 $front$ 等于队尾指针 $rear$ 时表示队空。若为 $front=11, rear=19$ ，则队列中的元素个数为 8。
5. A 是一个 10×10 的对称矩阵，若采用行优先的下三角压缩存储，第一个元素 $a[0][0]$ 的存储地址为 1，每个元素占一个存储单元，则 $a[7][5]$ 的地址为 26。
6. 在串定位运算中，模式串从目标的首位开始向右位移，每一次合法位移后如果模式串与目标中相应的字符相同，则这次位移就是有效位移。
7. 在具有 n 个结点的 k 叉树($k \geq 2$)的 k 叉链表表示中，有 $n(k-1)+1$ 个空指针。
8. 确定了排序的源点，DAG 图中无前趋顶点只有一个且从该点到终点只有一条路径时，它的拓扑序列才是唯一的。
9. 将二叉排序树 T 的先序序列中的关键字依次插入一空树中，所得和二叉排序树 T' 与 T 相同。
10. 快速排序的最大递归深度是 $\log_2 n + 1$ 。

二、 选择题（每小题 2 分，共 20 分）

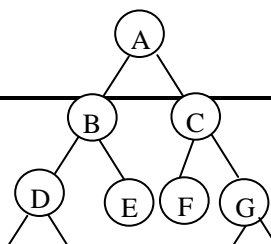
1. 算法在发生非法操作时可以作出处理的特性称为算法的_____。
A. 正确性 B. 易读性 C. 健壮性 D. 高效性
2. 在_____的运算中，使用顺序表比链表好。
A. 插入 B. 根据序号查找 C. 删除 D. 根据元素查找

3. 指针 p_1 和 p_2 分别指向两个无头结点的非空单循环链表中的尾结点，要将两个链表链接成一个新的单循环链表，应执行的操作为_____。
- A. $p_1 \rightarrow next = p_2 \rightarrow next; p_2 \rightarrow next = p_1 \rightarrow next;$
 B. $p_2 \rightarrow next = p_1 \rightarrow next; p_1 \rightarrow next = p_2 \rightarrow next;$
 C. $p = p_2 \rightarrow next; p_1 \rightarrow next = p; p_2 \rightarrow next = p_1 \rightarrow next;$
 D. $p = p_1 \rightarrow next; p_1 \rightarrow next = p_2 \rightarrow next; p_2 \rightarrow next = p;$
4. 设栈 S 和队列 Q 初始状态都为空，元素 $abcdefg$ 依次进入栈 S 。若每个元素出栈后立即进入队列 Q ，且 7 个元素出队的顺序是 $bdcfeag$ ，则栈 S 的容量至少是_____。
- A. 3 B. 2 C. 1 D. 其它
5. 下列广义表用图来表示时，分支结点最多的是_____。
- A. $A=(s,(a,B))$ B. $L=((x,(a,B)),(x,(a,B),y))$
 C. $B=((x,(a,B),y))$ D. $D=((a,B),(c,(a,B),D))$
6. 若字符串 "ABCDEFGH" 采用链式存储，假设每个指针占用 2 个字节，若希望存储密度 50%，则每个结点应存储_____个字符。
- A. 2 B. 3 C. 4 D. 5
7. 设一棵二叉树结点的先序遍历序历为：ABDGHCEFI，中序遍历序历为：GDHBAECIF，则二叉树中叶结点是_____。
- A. DEG B. EIGH C. BDHI D. 其它
8. 对于一个有向图，若一个顶点的度为 k_1 ，出度为 k_2 ，则对应邻接表中该顶点单链表中的边结点数为_____。
- A. k_1 B. k_2 C. $k_1 - k_2$ D. $k_1 + k_2$
9. 下列关于 m 阶 B-树的叙述中，错误的是_____。
- A. 每个结点至多有 m 个关键字
 B. 每个结点至多有 m 棵子树
 C. 插入关键字时，通过结点分裂使树高增加
 D. 删除关键字时通过结点合并使树高降低
10. 用直接插入排序法对下面的四个序列进行由小到大的排序，元素比较次数最少的是_____。
- A. 94, 32, 40, 90, 80, 46, 21, 69 B. 21, 32, 46, 40, 80, 69, 90, 94
 C. 32, 40, 21, 46, 69, 94, 90, 80 D. 90, 69, 80, 46, 21, 32, 94, 40

三、 应用题（每小题 10 分，共 50 分）

1. 设题 3-1 图所示二叉树是与某森林对应的二叉树，试回答下列问题。

(1) 森林中有几棵树？



-
- (2) 每一棵树的根结点分别是什么？
 - (3) 第一棵树有几个结点？
 - (4) 第二棵树有几个结点？
 - (5) 森林中有几个叶结点？

题 3-1 图

- 2. 给定一个权集 $W=\{4, 5, 7, 8, 6, 12, 18\}$ ，试画出相应的哈夫曼树，并计算其带权路径长度 WPL。
- 3. 已知含 8 个顶点 ABCDEFGH 的连通带权图的邻接矩阵如题 3-3 图所示，试画出它的最小生成树（设起点为 A），问最小生成树的权值和为多少？

$$\begin{bmatrix} 0 & 4 & 3 & \infty & \infty & \infty & \infty & \infty \\ 4 & 0 & 5 & 5 & 9 & \infty & \infty & \infty \\ 3 & 5 & 0 & 5 & \infty & \infty & \infty & 5 \\ \infty & 5 & 5 & 0 & 7 & 6 & 5 & 4 \\ \infty & 9 & \infty & 7 & 0 & 3 & \infty & \infty \\ \infty & \infty & \infty & 6 & 3 & 0 & 2 & \infty \\ \infty & \infty & \infty & 5 & \infty & 2 & 0 & 6 \\ \infty & \infty & 5 & 4 & \infty & \infty & 6 & 0 \end{bmatrix}$$

题 3-3 图

- 4. 对于给定结点的关键字集合 $K=\{34, 76, 45, 18, 26, 54, 92, 38\}$ ，
 - (1) 试构造一棵二叉排序树；
 - (2) 求等概率情况下的平均查找长度 ASL。
- 5. 对关键字序列 (429, 653, 275, 897, 170, 908, 473, 256, 726) 进行 LSD 基数排序，写出每一趟的排序结果。

四、 算法题（每小题 10 分，共 20 分）

- 1. 链栈的存储结构和栈顶指针定义如下，f41 算法的功能是把十进制数转换为二进制数。试完成下列程序填空。

```
#define MAXLEN 100
typedef struct stacknode                                // 定义栈的存储结构
{ int data;
  struct stacknode *next;
```

```

}stacknode;
typedef struct
{ stacknode  *top;                                // 定义栈顶的指针
}linkstack;

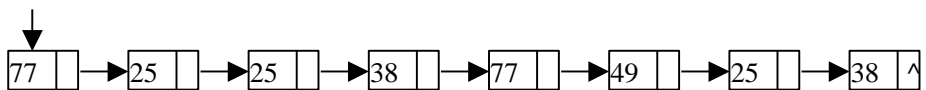
void f41(int n)                                    // 栈的应用：二—十进制转换
{ linkstack s;
  int x;
  s.top=NULL;                                     // 置栈空
  do
  {   x=n%2;
      n=_____①_____ ;
      stacknode *p=new _____②_____ ;
      p->next=s.top ;
      s.top=p;
      s.top->data=_____③_____ ;
  }
  while (n);
  cout<< " 转换后的二进制数值为:  ";
  while (s.top)
  {   cout<< _____④_____ ;
      stacknode *p=s.top;
      s.top= _____⑤_____ ;
      delete  p ;
  }
}

```

2. 已知一个不带头结点的单链表 **head**，结点结构为 **Node**。写出一个通用的算法，删除链表中值相同的结点（如果有值相同的结点只保留首次出现的结点）。

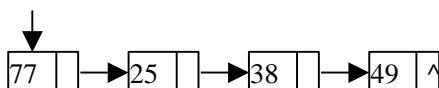
例如，删除前：

head



删除后：

head



预编译命令：

```
# define  NULL  0  //NULL 表示空指针
```

结点结构：

```
typedef  struct node{  
    int  data;  
    struct node  *next;  
}Node;
```

函数首部：

```
Node  * f42 (Node  * head)
```