

1. 对于插入、删除而言，线性表的顺序存储优于链式存储。

2. 栈和队列是操作上受限制的线性表。

3. 假设 B 是一棵树，B' 是对应的二叉树。则 B 的后根遍历相当于 B' 的中序遍历。

4. 通常，深度为 K 的二叉树有  $2^k-1$  个结点。

5. 一个无向连通图的生成树是其极小的连通子图。

6. 关键路径是 AOE 网中源点到汇点的最长路径。

7. 对于一棵 m 阶的 B 树，每个除根外的非终端结点至少有  $\lceil m/2 \rceil-1$  关键字。至多有 m 棵子树。

8. 对于具有 N 个结点的二叉排序树而言，查找长度任何情况下都与  $O(\log n)$  同阶。

9. 对于任何待排序序列来说，快速排序都是最快的。

10. 选择类排序是不稳定的。

二. 选择题（每选 2 分共 20 分）

1. 深度优先遍历图使用了数据结构（ ），而广度优先遍历图使用了数据结构（ ）。

a. 数组                      b. 栈                      c. 队列                      d. 线性

2. 若广义表 LS 满足 Head(LS)==Tail(LS)，则 LS 为( )。

a. ( )                      b. (( ))                      c. (( ), ( ))                      d. (( ), ( ), ( ))

3. 若入栈顺序为 1、2、3、4，则下列( )的出栈序列是不可能的。

a. 1、2、3、4                      b. 4、3、2、1                      c. 4、1、2、3                      d. 3、2、1、4

4. 栈和队列的一个共同点是( )。

a. 都是先进后出                      b. 都是先进先出                      c. 只允许在端点处插入和删除元素                      d. 没有共同点

5. 设森林 F 中有三棵树，第一、第二和第三棵树的结点个数分别为 m1、m2 和 m3，则与森林 F 对应的二叉树的右子树中有（ ）个结点。

a. m1                      b. m1+m2                      c. m3                      d. m2+m3

6. 下列算法中（ ）适用于求图的单源最短路径。

a. DFS 算法                      b. BFS 算法                      c. Prim 算法                      d. Dijkstra 算法

7. 在待排序序列“基本有序”情况下，较好的内部排序方法是( )。

a. 简单选择排序                      b. 直接插入排序                      c. 归并排序                      d. 快速排序

8. 静态查找表与动态查找表的根本区别在于( )。

a. 它们的逻辑结构不同                      b. 施加在其上的操作不同                      c. 所包含的数据元素的类型不同                      d. 存储实现不同

三. 填空题（每空 2 分共 20 分）

1. 在一棵高度为 4 的平衡二叉树中，至少有（ ）个结点，至多有（ ）个结点。

2. 在一棵深度为 k 的二叉树中( $k \geq 1$ )，所有非终端结点都有左、右子树，且非终端结点有  $2^{k-1}$  个，则该二叉树的终端结点有（ ）个。该二叉树称为（ ）二叉树。

3. 若有序表中关键字序列为：15，20，25，30，35，40，45，50，55，60，65，70。对其进行折半查找，则在等概率情况下，查找成功时的平均查找长度是（ ）。查找关键字 20 时需进行（ ）次比较。

4. 已知 Hash 函数为  $H(K) = K \bmod 13$ ，若用平方（二次）探测再散列处理冲突，关键字（23，34，56，24，75，12，49，52）散列在地址 0 -14 后，关键字 49 的地址为（ ）。该表平均成功的查找长度为（ ）、

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

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

5. 假设用于通讯的电文仅由 6 个字符组成，字母在电文中出现的频率分别为 15，6，32，18，21，8。若为这 6 个字母设计哈夫曼编码（设生成新的二叉树的规则是按给出的次序从左至右的结合，新生成的二叉树总是插入在最右），则频率为 8 的字符编码是（ ），频率为 21 的字符编码是（ ）。

四. 结构问答题（每题 6 分共 24 分）

1. 已知某二叉树的先序次序为 ABDECFG，中序次序为 DEBAFCG，试构造该二叉树，并为之建立中序线索。

2. 对如右所示的有向图，

a. 画出其邻接表；

b. 针对你所建的邻接表，写出该图的

深度优先遍历序列和广度优先遍历序列

A

C

6

b

d

7

5

8

A

C

6

b

d

7

5

8

3. 对关键字序列（5,10,25,20,30,15,4,25），构造一棵平衡的二叉排序树，请画出每加入一个新结点后的结果。若发生不平衡,指明需做的平衡旋转类型。

4. 已知待排序的一组记录关键字的初始排列如下：37,24,42,6,53,8,72,11,3,9。若按关键字递增有序排序，请给出：

a. 快速排序完成第一趟划分之后的记录排列序列；

b. 堆排序初始建堆（大顶堆）的序列；

c. 第一趟基数排序后的序列。

五. 算法设计题（共 16 分）

1. 已知某线性表以带头结点的单链表表示，结点结构为：

typedef struct LNode {

ElemType data;

struct Lnode \*next;

} LNode, \*LinkList;

写一算法 Oplinklist(linklist L,int i;int j)

删除线性表中第 i 个元素,并将之插入至原表中的第 j 个元素之前。

2. 设树 T 用孩子-兄弟法表示。

typedef struct CSNode {

ElemType data;

Struct CSNode \*firstchild, \*nextsibling;

} CSNode, \*CSTree;

请编写求树中叶子结点个数的算法。（8 分）

第 1 页                      共 1 页