

一、单项选择题：(1-10 小题，每小题 1 分，共 10 分。在每小题给出的四个选项中，请选出一项最符合题目要求的。)

1. 设  $n$  是描述问题规模的非负整数，下面程序片段的时间复杂度是 ( )。

```
x=n*n;
while (x>=1)
    x=x/2;
```

A .  $O(\log_2 n)$       B .  $O(n)$       C .  $O(n\log_2 n)$       D .  $O(n^2)$

2. 需要分配一个较大的存储空间，并且插入和删除操作不需要移动元素，满足以上特点的线性表存储结构是 ( )。

A . 单向链表      B . 静态链表      C . 线性链表      D . 顺序表

3. 已知字符串  $S$  为“ababcabcacbab”，模式串  $T$  为“abcac”。若采用 KMP 算法进行模式匹配，则需要 ( ) 遍 (趟) 匹配，就能确定  $T$  是  $S$  的子串。

A . 3      B . 4      C . 5      D . 6

4. 已知某棵二叉树的前序序列是 1, 2, 3, 4, 则不可能为该二叉树的中序序列的是 ( )。

A . 1, 2, 3, 4      B . 2, 3, 4, 1      C . 1, 4, 3, 2      D . 3, 1, 4, 2

5. 将森林  $F$  转换为对应的二叉树  $T$ ,  $F$  中任何一个没有右兄弟的结点，在  $T$  中 ( )。

A . 没有左子树      B . 没有右子树  
C . 没有左子树和右子树      D . 以上都不对

6. 一个含有  $n$  个顶点和  $e$  条边的无向图，在其邻接矩阵存储结构中共有 ( ) 个零元素。

A .  $e$       B .  $2e$       C .  $n^2-2e$       D .  $n^2-e$

7. 在一棵高度为 2 的 7 阶 B 树中，所含关键字的个数最少是 ( )。

A . 5      B . 7      C . 8      D . 14

8. 设待排序的元素个数为  $n$ , 则基于比较的排序算法在最坏情况下的时间复杂度的下界为 ( )。

A .  $\log_2 n$       B .  $n$       C .  $n\log_2 n$       D .  $n^2$

9. 下面关于 B 树和 B+树的叙述中，不正确的是 ( )。

A . B 树和 B+树都能有效地支持随机检索      B . B 树和 B+树都能有效地支持顺序检索  
C . B 树和 B+树都是平衡的多路树      D . B 树和 B+树都可以用于文件的索引结构

10. 对于一组权值都相等的 16 个字母，构造相应的哈夫曼 (Huffman) 树，这棵哈夫曼树是 ( )。

A . 完全二叉树      B . 一般二叉树      C . 满二叉树      D . 二叉查找树

二、填空题：(11-17 小题，每空 1 分，共 10 分。)

11. 在一棵  $n$  个结点的二叉树中，所有结点的空子树个数为 ( )。

12. 分块查找是介于 ( 折半 (二分) 查找 ) 和 ( 顺序 (线性、直接) 查找 ) 之间的查找方法。对  $n$  个元素进行分块查找，平均每块长度为 (  $\sqrt{n}$  ) 时效率最好高。

13.  $n$  个元素进行二路归并排序，需要归并  $(\log_2 n)$  次，算法时间复杂度为 (  $n\log_2 n$  (大 O 写不写均可) )。

14.  $n$  个顶点  $e$  条边的无环路有向图，若采用邻接表作为存储结构，则拓扑排序算法的时间复杂度为 (  $O(n+e)$  )。

15. 加权无向图的最小生成树的特点是 (  $n$  个顶点，  $n-1$  条边，权值和最小 )。

16. 在具有 12 个结点的平衡二叉树 (AVL 树) 中，查找 AVL 树中的一个关键字最多需要 ( 5 ) 次比较。

17. 带权有向图的一条关键路径是指 ( 从源点到汇点路径长度之和最大的 (回答定义也

对))。

### 三、简答题：(共 25 分)

1. (8 分) 在  $n$  个数据中找出前  $K$  个最大元素，可以采用堆排序思想或败者树思想来实现。分别说明上述两种实现方法的基本步骤，并分析每种方法的时间复杂度和空间占用情况。

**参考答案：**

维护  $K$  个元素的大根堆，读入新数据，比根大则替换根结点，再整理成堆，比根小，则跳过，直到所有元素都处理完。时间复杂度  $O(n \log k)$ ，空间使用  $K$ ；败者树用  $K$  各元素作为叶节点，时间也相同，空间使用为  $2K-1$ 。

评分标准：思想各 3 分，时空各 1 分

也可以建立大根堆，利用堆排序，取堆顶  $K$  次即可。时间复杂度为  $O(n \log n)$ ，空间为  $N$ ；败者树用  $N$  各元素作为叶节点，时间也相同，空间使用为  $2N-1$ 。

2. (10 分) 假设举办一个 1000 人参加的学术会议。作为会议报到组的负责人，你会收到会务组为每名参会者开具的包含其英文名字的注册费发票，同时还会收到为每位参会者提供的印有其英文名字的参会胸牌和其他会议资料。请回答以下问题：

(1) 如何有效地把每个参会者的注册费发票和参会胸牌等其他会议资料放在一起形成一份参会资料？

(2) 如何在会议报到日更有效地把每份参会资料发放给参会者？

要求：说明你所使用的主要技术和关键步骤。

**参考答案：**

主要技术（写出散列、分块、分组、桶、基数等思想 4 分）；

关键步骤 (1)、(2) 各 3 分。写出散列函数或分块方法等划分方法，写出查找方案。

3. (7 分) 设某文件经预处理后，得到 11 个初始归并段，归并段中的每个数据占一个磁盘读写单位，归并段长度分别为： $\{47, 9, 39, 18, 4, 12, 23, 7, 21, 16, 26\}$ 。假设采用 4 路归并，请设计一个读写文件磁盘次数最少的归并方案。要求：画出最佳归并树，并计算磁盘的读写次数。

### 四、算法设计题：(共 25 分)

按以下要求设计算法：

(1) 给出算法的基本设计思想。

(2) 使用 C 或 C++ 或 Java 语言，给出相关的数据类型定义。

(3) 根据设计思想，采用 C 或 C++ 或 Java 语言描述算法，关键之处给出注释。

(4) 说明你所设计算法的时间复杂度。

1. (8 分) 设计一种数据结构，满足队列的“先进先出”性质，实现以下 3 种操作：

1) Enqueue( $v$ )：将  $v$  加入到队列；

2) Dequeue()：删除队首元素并返回此元素；

3) Maxelement()：返回队列中最大元素；

并且使每种操作的时间复杂度尽可能的低。

**算法思想：**可以用两个栈模拟队列操作，多设一个栈，用于存放最大值，入队复杂度为  $O(1)$ ，出队为  $O(\text{size}(\text{Stack}))$ ，求最大值为  $O(1)$

2. (10 分) 给定一棵  $n$  个结点的二叉排序树（即 BST），每个结点均存放一个整数，其结点格式为  $[\text{lchild}][\text{data}][\text{rchild}]$ 。令  $\text{half} = (\text{BST 中的最大值} + \text{BST 中的最小值}) / 2$ 。设计一个算法  $\text{int findNearMid}(\text{BinTree} * \text{root})$ ，完成：(1) 找出 BST 中最大和最小值以计算  $\text{half}$  的值；(2) 返回大于  $\text{half}$  且与  $\text{half}$  相差最小的结点值。

评分标准：基本思想 4 分，数据结构 2 分，代码 6 分，时间复杂度 1 分，没有思想，代码正确不扣分

参考答案：

算法思想：在 BST 中，最小值就是最左边的结点，最大值就是最右边的结点。在分别求出 min 和 max 后，求出 half。然后利用查找，找出第一个大于 half 的结点即为所求。 $O(\log n)$  或中序遍历最左最右即为所求，顺序或折半找出大于 half 的结点，复杂度为  $O(n)$ 。

数据结构：

```
struct Node{
    int data;
    Node *lchild;
    Node *rchild;
} *BinTree;

int FindNode(BinTree root)
{
    Node * ptr = *root;
    int min, max;
    while (ptr != NULL)
    {
        min = ptr->data;
        ptr = ptr->lchild;
    }
    ptr = *root;
    while (ptr != NULL)
    {
        max = ptr->data;
        ptr = ptr->rchild;
    }
    int half = (min + max) >> 1;
    ptr = *root;
    while (ptr != NULL)
    {
        if (ptr->data < half)
            ptr = ptr->rchild;
        else if (ptr->data > half) return ptr->data;
        else return (ptr->rchild)->data;
    }
}
```

3 . (7 分) 某加权无环路有向图各边权值均相等，采用邻接矩阵存储，设计一个算法 int FindLpath( G , u, v)，求出任意两点 u, v 的最长距离及路径。

算法思想：可以使用 Floyd 算法，改  $A[i][k] + A[k][j] > A[i][j]$ ，即可；也可以是由广度搜索，计算路径长度，并保存前导结点即可。