

## 一、选择题

下面关于串的叙述中, 哪一个是不正确的? ( )

- A . 串是字符的有限序列                      B . 空串是由空格构成的串  
C . 模式匹配是串的一种重要运算    D . 串既可以采用顺序存储, 也可以采用链式存储

设无向图的顶点个数为 n, 则该图最多有 ( ) 条边。

- A . n-1        B . n(n-1)/2        C . n(n+1)/2    D . 0

以下数据结构中, ( ) 是非线性数据结构。

- A . 树            B . 字符串        C . 队列            D . 栈

下面关于线性表的叙述中, 错误的是哪一个? ( )

- A . 线性表采用顺序存储, 必须占用一片连续的存储单元。  
B . 线性表采用顺序存储, 便于进行插入和删除操作。  
C . 线性表采用链接存储, 不必占用一片连续的存储单元。  
D . 线性表采用链接存储, 便于插入和删除操作。

假设以数组 A[m]存放循环队列的元素,其头尾指针分别为 front 和 rear, 则当前队列中的元素个数为 ( )。

- A . (rear-front+m)%m                      B . rear-front+1  
C . (front-rear+m)%m                      D . (rear-front)%m

在单链表指针为 p 的结点之后插入指针为 s 的结点,正确的操作是 ( )。

- A . p->next=s; s->next=p->next;    B . s->next=p->next; p->next=s;  
C . p->next=s; p->next=s->next;    D . p->next=s->next; p->next=s;

设栈的输入序列是 1, 2, 3, 4,则 ( ) 不可能是其出栈序列。

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

栈和队都是 ( )

- A . 顺序存储的线性结构                      B . 链式存储的非线性结构  
C . 限制存取点的线性结构                      D . 限制存取点的非线性结构

从逻辑上可以把数据结构分为 ( ) 两大类。

- A . 动态结构、静态结构    B . 顺序结构、链式结构  
C . 线性结构、非线性结构    D . 初等结构、构造型结构

下列四个序列中, 哪一个 是堆 ( )。

- A . 75,65,30,15,25,45,20,10        B . 75,65,45,10,30,25,20,15

- C . 75,45,65,30,15,25,20,10        D . 75,45,65,10,25,30,20,15

在下述结论中, 正确的是 ( )

- ①只有一个结点的二叉树的度为 0;  
②二叉树的度为 2;  
③二叉树的左右子树可任意交换;  
④深度为 K 的完全二叉树结点个数小于或等于深度相同的满二叉树。

- A . ①②③            B . ②③④            C . ②④            D . ①④

若一棵二叉树具有 10 个度为 2 的结点, 5 个度为 1 的结点, 则度为 0 的结点个数是 ( )

- A . 9                      B . 11                      C . 15                      D . 不确定

(n0=n2+1)

在下面的程序段中, 对 x 的赋值语句的频率为 ( )。

```
for ( i=1;i< n;i++)  
    for(j=1;j<ij++)  
        x=x+1;
```

- A . O(2n)        B . O(n)        C . O(n2)        D . O(log2n)

一个 n 个顶点的连通无向图, 其边的个数至少为 ( )。

- A . n-1            B . n            C . n+1            D . nlogn;

根结点在第 1 层的二叉树的第 l 层上最多含有结点数为 ( )

- A . 2<sup>l</sup>            B . 2<sup>^(l-1)</sup>-1            C . 2<sup>^(l-1)</sup>            D . 2<sup>l</sup>-1

下列排序算法中( C )排序在一趟结束后不一定能选出一个元素放在其最终位置上。

- A . 选择            B . 冒泡            C . 归并            D . 堆

散列文件使用散列函数将记录的关键字值计算转化为记录的存放地址, 因为散列函数是一一对应的关系, 则选择好的 ( ) 方法是散列文件的关键。

- A . 散列函数                                      B . 除余法中的质数  
C . 冲突处理                                      D . 散列函数和冲突处理

在一个长度为 n 的顺序表中, 向第 i 个元素(1≤i≤n+1)之前插入一个新元素时, 需向后移动 ( ) 个元素。

- A. n-1                      B. n-i+1                      C. n-i-1                      D. i

在一个具有 n 个单元的顺序栈中, 假定以地址低端作为栈底, 以 top 作为栈顶指针, 则当做退栈处理时, top 变化为 ( )。

- A. top 不变          B. top = -n          C. top = top-1    D. top=top+1

向顺序栈中压入元素时，是（    ）。

- A. 先存入元素，后移动栈顶指针      B. 先移动栈顶指针，后存入元素

说明：若顺序栈 top 初始值为-1，则选 B，若初始值为 0，则可选 A

在一个顺序存储的循环队列中，队首指针指向队首元素的（    ）。

- A. 前一个位置      B. 后一个位置      C. 队首元素位置      D. 队尾元素位置

若进栈序列为 1,2,3,4,进栈过程中可以出栈，则（    ）不可能是一个出栈序列。

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

在具有 n 个单元的顺序存储的循环队列中，假定 front 和 rear 分别为队首指针和队尾指针，则判断队空的条件是（    ）。

- A. front= =rear+1    B. front+1= =rear    C. front= =rear      D. front= =0

在具有 n 个单元的顺序存储的循环队列中，假定 front 和 rear 分别为队首指针和队尾指针，则判断队满的条件是（    ）。

- A. rear % n= =front      B. (rear-1) % n= =front

- C. (rear-1) % n= =rear    D. (rear+1) % n= =front

从一个具有 n 个节点的单链表中查找其值等于 x 结点时，在查找成功的情况下，需平均比较（    ）个结点。

- A. n                  B. n/2                  C. (n-1)/2            D. (n+1)/2

在一个单链表中，已知\*q 结点是\*p 结点的前驱结点，若在\*q 和\*p 之间插入\*s 结点，则执行（    ）。

- A. s->next=p->next; p->next=s;          B. p->next=s->next; s->next=p;

- C. q->next=s; s->next=p;                  D. p->next=s; s->next=q;

向一个带头结点的链栈中插入一个\*s 结点时，其中栈顶指针为 hs，则执行（    ）。

- A. hs->next=s;                          B. s->next=hs->next;    hs->next=s;

- C. s->next=hs;    hs=s;                  D. s->next=hs; hs=hs->next;

注：若不带头结点，C 也可以

在一个链队列中，假定 front 和 rear 分别为队首指针和队尾指针，则进行插入\*s 结点的操作时应执行（    ）。

- A. front->next=s; front=s;                  B. rear->next=s; rear=s;

- C. front=front->next;                          D. front=rear->next;

线性表是（    ）。

- A. 一个有限序列，可以为空                  B. 一个有限序列，不能为空

- C. 一个无限序列，可以为空                  D. 一个无限序列，不能为空

对顺序存储的线性表，设其长度为 n，在任何位置上插入或删除操作都是等概率的，删除一个元素时大约要移动表中的（    ）个元素。

- A. n+1                  B. n-1                  C. (n-1)/2              D. n

线性表采用链式存储时，其地址（    D ）。

- A. 必须是连续的                          B. 部分地址必须是连续的

- C. 一定是不连续的                          D. 连续与否均可以

设单链表中指针 p 指着结点(数据域为 m)，指针 f 指着将要插入的新结点(数据域为 x)，当 x 插在结点 m 之后时，要先修改（    ）后修改 p->link=f 即可。

- A. f->link=p;                                  B. f->link=p->link;

- C. p->link=f->link;                          D. f=nil;

根据线性表的链式存储结构，每个结点所含指针的个数，链表分为单链表和（    ）。

- A. 循环链表      B. 多重链表      C. 普通链表      D. 无头结点链表

在数据结构中，与所使用的计算机无关的数据叫（    ）结构。

- A. 存储                  B. 物理                  C. 逻辑                  D. 物理和存储

二分法查找（    ）存储结构。

- A. 只适用于顺序      B. 只适用于链式      C. 既适用于顺序也适用于链式

- D. 既不适合于顺序也不适合于链式

在线性表的链式存储结构中，逻辑上相邻的元素在物理位置上（    ）。

- A. 一定相邻                  B. 不一定相邻                  C. 有时相邻

设字符串 s1='abcdefg',s2='pqrst'，则运算

s=concat(sub(s1,2,len(s2)),sub(s1,len(s2),2))后串值为（    ）。

- 'bcdef'                  B. 'bcdefg'                  C. 'bcpqrst'                  D. 'bcdefef'

注：concat 连接字符串，sub 求子串

假定在一棵二叉树中，双分支结点数为 15 个，单分支结点数为 32 个，则叶子结点数为（    ）。

- A. 15                  B. 16                  C. 17                  D. 47

假定一棵二叉树的结点数为 18 个，则它的最小高度（    ）。

- A. 4                  B. 5                  C. 6                  D. 18

在一棵二叉树（根结点所在的层为 1）中第五层上的结点数最多为（    ）。

- A. 8                      B. 15                      C. 16                      D. 32

在一棵具有五层的满二叉树中，结点总数为（ ）。

- A. 31                      B. 32                      C. 33                      D. 16

已知 8 个数据元素为(34、76、45、18、26、54、92、65)，按照依次插入结点的方法生成一棵二叉排序树后，最后两层上的结点总数为（ ）。

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

由分别带权为 9、2、5、7 的四个叶子结点构造一棵哈夫曼树，该树的带权路径长度为（ ）。

- A. 23                      B. 37                      C. 44                      D. 46

下面答案（ ）是查找二叉树（又称二叉排序树）。

- A. 二叉树中的每个结点的两棵子树的高度差的绝对值不大于 1  
B. 二叉树中的每个结点的两棵子树的高度差等于 1  
C. 二叉树中的每个结点的两棵子树是有序的  
D. 二叉树中的每个结点的关键字大于其左子树（如果存在）所有结点的关键字值，且小于其右子树（如果存在）所有结点的关键字值。

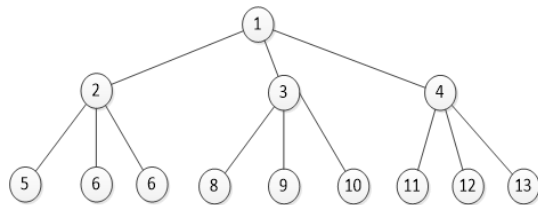
如果结点 A 有三个兄弟，而且 B 是 A 的双亲，则 B 的出度是（ ）。

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

一个深度为 L 的满 k 叉树有如下性质：第 L 层上的结点都是叶子结点，其余各层上每个结点都有 k 棵非空子树。如果按层次顺序从 1 开始对全部结点编号，编号为 n 的有右兄弟的条件是（ ）。

- (n-1) % k = 0    B. (n-1) % k != 0    C. n % k = 0    D. n % k != 0

注：以满三叉树为例



在完全二叉树（结点编号从 1 开始）中，当 i 为奇数且不等于 1 时，结点 i 的左兄弟是结点（ ），否则没有左兄弟。

- A. 2i-1                      B. i+1                      C. 2i+1                      D. i-1

在一个有向图中，所有顶点的入度之和等于所有顶点的出度之和的（ ）倍。

- A. 1/2                      B. 1                      C. 2                      D. 4

对于一个具有 n 个顶点和 e 条边的无向图，若采用邻接表表示，则表头向量的大小为（ ）。

- A. n                      B. n+1                      C. n-1                      D. n+e

具有 n 个顶点的无向完全图，边的总数为（ ）条。

- A. n-1                      B. n                      C. n+1                      D. n\*(n-1)/2

设图 G 有 n 个顶点和 e 条边，当 G 是非孤立顶点的连通图时有  $2e \geq n$ ，故可推得深度优先搜索的时间复杂度为（ ）。（最坏的情况就是把所有的边都遍历了一遍）

- A. O(e)                      B. O(n)                      C. O(ne)                      D. O(n+e)

最小代价生成树（ ）。

- A. 是唯一的    B. 不一定是唯一的，但最小代价是唯一的。

在无向图 G 的邻接矩阵 A 中，若  $A[i,j]$  等于 1，则  $A[j,i]$  等于（ ）。

- A. i+j                      B. i-j                      C. 1                      D. 0

对于长度为 9 的顺序存储的有序表，若采用折半查找，在等概率情况下的平均查找长度为（ ）。

- A. 20/9                      B. 18/9                      C. 25/9                      D. 22/9

已知一个有序表为(12、18、24、35、47、50、62、83、90、115、134)，当二分查找值为 90 的元素时，（ ）次比较后查找成功；当二分查找值为 47 的元素时，（ ）次比较后查找成功。

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

散列函数有一个共同性质，即函数值应当以（ ）取其值域中的每个值。

- A. 最大概率                      B. 最小概率                      C. 平均概率                      D. 同等概率

设散列地址空间为  $0 \sim m-1$ ，k 为关键字，用 p 去除 k，将所得的余数作为 k 的散列地址，即  $H(k) = k \% p$ 。为了减少发生冲突的频率，一般取 p 为（ ）。

- A. 小于 m 的最大奇数                      B. 小于 m 的最大偶数  
C. m                      D. 小于等于 m 的最大素数

设关键字序列为：3,7,6,9,8,1,4,5,2。进行简单选择排序的最少交换次数是（ ）。

- A. 6                      B. 7                      C. 8                      D. 20

一组记录排序码为(46、79、56、38、40、84)，则利用堆排序的方法建立的初始堆为（ ）。

- A. (79、46、56、38、40、80)                      B. (84、79、56、38、40、46)  
C. (84、79、56、46、40、38)                      D. (84、56、79、40、46、38)

一组记录的关键码为(46、79、56、38、40、84)，则利用快速排序的方法，以第一个记录

为基准得到的一次划分结果为（ ）。

- A. (38、40、46、56、79、84)
- B. (40、38、46、79、56、84)
- C. (40、38、46、56、79、84)
- D. (40、38、46、84、56、79)

二、填空题

在线性结构中第一结点 前驱结点，其余每个结点有且只有 个前驱结点；最后一个结点 后继结点，其余每个结点有且只有 个后继结点。

在树型结构中，树根结点没有 结点，其余每个结点有且仅有 个父亲结点；树叶结点没有 结点，其余每个结点的 结点数不受限制。

广义表(((a,b),c),d,(c))的表头 (head) 是 ；表尾 (tail) 是 。

一个 n 阶下三角矩阵 A 按行优先顺序压缩存放在一维数组 B 中，则 B 中的元素个数是 (需要一个元素，存储非下三角元素的值)

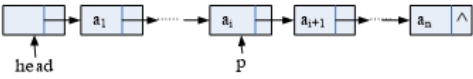
对于长度为 n 的顺序表，插入或删除元素的时间复杂性为 O ；对于顺序栈或队列，插入或删除元素的时间复杂性为 。

设有一顺序栈 S，元素 s1,s2,s3,s4,s5,s6 依次进栈，如果 6 个元素出栈的顺序是 s2,s3,s4, s6 , s5,s1,则栈的容量至少应该是 。

在线性表的顺序存储中，元素之间的逻辑关系是通过 决定的；在线性表的链接存储中，元素之间的逻辑关系是通过 决定的。

一棵结点总数为 9 的完全二叉树，有 个叶子结点。

单链表如下图，完成对它的基本操作



删除\*p 结点时，应执行如下操作:

- (1)q=p->next;
- (2)p->data=p->next->data;
- (3)p->next= ;
- (4)free(q);

若要在一个单链表中的\*p 结点之前插入一个\*s 结点时，可执行如下操作

- (1)s->next= ;
- (2)p->next=s;
- (3)t=p->data;

(4)p->data= ;

(5)s->data= ;

无论对于顺序存储还是链接存储的栈和队列来说，进行插入或删除运算的时间复杂性均相同，则为 。

$\begin{bmatrix} 0 & 0 & 2 & 0 \end{bmatrix}$

一个稀疏矩阵为  $\begin{bmatrix} 3 & 0 & 0 & 0 \end{bmatrix}$ ，则对应的三元组线性表为

$\begin{bmatrix} 0 & 0 & -1 & 5 \end{bmatrix}$

$\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$

对于一棵具有 n 个结点的树，则该树中所有结点的度之和为 。

在一棵二叉树中，度为 0 的结点的个数为 n0，度为 2 的结点的个数为 n2，则 n0 = 。

在二叉树的顺序存储中(下标从 1 开始)，对于下标为 5 的结点，它的双亲结点的下标为 ，若它存在左孩子，则左孩子结点的下标为 ，若它存在右孩子，则右孩子结点的下标为 。

若广义表表示为(b(c,d(e,f,g(h)),i(j,k(x,y))),则广义表的长度为 ，深度为 (假设根结点的层数为 1) (长度的求法为最大括号中的逗号数加 1，深度的求法为最大括号中每个元素的括号匹配数加 1 的最大值)

假定一棵二叉树的广义表表示为 A(B(, D), C(E(G, F))),则该树的深度为 ，度为 0 的结点数为 ，度为 1 的结点数为 ，度为 2 的结点数为 ；C 结点是 A 结点的 孩子，E 结点是 C 结点的 孩子。

在一棵二叉排序树中，按 遍历得到的结点序列是一个有序序列。

由分别带权为 3,9,6,2,5 的共五个叶子结点构成一棵哈夫曼树，则带权路径长度为 。

假定在二叉树的链接存储中，每个结点的结构为 left data right

data 为值域，left 和 right 分别为链接左、右孩子结点的指针域，请在下面中序遍历算法中画有横线的地方填写合适的语句。

```
void inorder(node *bt) (bt 是指向结点的指针)
{
    if (bt!=null ) {
        (1) ;
        (2) ;
        (3) ; }
}
```

在图 G 的邻接表表示中，每个顶点邻接表中所含的结点数，对于无向图来说等于该顶点的 ，对于有向图来说等于该顶点的 。

假定一个无向连通网图具有 n 个顶点和 e 条边，则采用邻接矩阵表示的空间复杂性为                     ，采用邻接表表示的空间复杂性为                     ，采用 Prim 算法求最小生成树的时间复杂度是                     ，采用克鲁斯卡尔算法求最小生成树的时间复杂度是                     。

已知一个无向图的邻接矩阵如下所示，则从顶点 A 出发按深度优先搜索遍历得到的顶点序列为                     ，按广度优先搜索遍历得到的顶点序列为                     。

	A	B	C	D	E	F
A	0	1	1	0	1	0
B	1	0	1	0	1	1
C	1	1	0	1	0	0
D	0	0	1	0	0	1
E	1	1	0	0	0	1
F	0	1	0	1	1	0

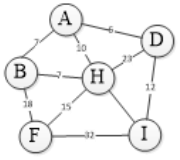
在索引查找或分块查找中，首先查找                     ，然后再查找相应的                     ，整个索引查找的平均查找长度等于查找索引表的平均查找长度与查找相应子表的平均查找长度之                     。

在散列存储中，装填因子α的值越大，存取元素时发生冲突的可能性就                     ，当 α 的值越小，存取元素时发生冲突的可能性就                     。

给定线性表(18,25,63,50,42,32,90)，用散列方式存储，若选用 h(K)=K % 9 作为散列函数，则元素 18 的同义词元素共有              个，元素 25 的同义词元素共有              个，元素 50 的同义词元素共有              个。

在对一组记录(54,38,96,23,15,72,60,45,83)进行直接选择排序时，第四次选择和交换后，未排序记录(即无序表)为                     。

已知一个图如下所示，在该图的最小生成树的代价为                     。



给出从顶点 1 到顶点 8 的关键路径长度                     。（最长路径）

假定在有序表 A[1..20]上进行二分查找，则比较一次查找成功的结点数为                     ，比较两次查找成功的结点数为                     ，比较三次查找成功的结点数为                     ，比较四次查找成功结点数为                     ，比较五次查找成功的结点数为                     ，平均查找长度为                     。

在对一组记录(54,38,96,23,15,72,60,45,38)进行冒泡排序时，第一趟需进行相邻记录交换的次数为             ，在整个冒泡排序过程中共需进行              趟后才能完成。

注：这里是从后往前冒泡的结果

在归并排序中，若待排序记录的个数为 20，则共需要进行              趟归并，在第三趟归并中，是把长度为              的有序表归并为长度为              的有序表。

在直接插入和简单选择排序中，若初始数据基本正序，则选用                     ，若初始数据基本反序，则选用                     。

在堆排序、快速排序和归并排序中，若只从节省空间考虑，则应首先选取                      方法，其次选取                      方法，最后选取                      方法；若只从排序结果的稳定性考虑，则应选取                     ；若只从平均情况下排序最快考虑，则应选取                      方法；若只从最坏情况下排序最快并且要节省内存考虑，则应选取                      方法。

三、判断题

数据元素是数据的最小单位（ X ）。（数据项是最小单位，数据元素是基本单位）

数据项是数据的基本单位（ X ）。

顺序存储的线性表可以随机存取（ √ ）。

线性表中的元素可以是各种各样的，但同一线性表中的数据元素具有相同的特性，因此，是属于同一数据对象（√ ）。

在单链表中，任何两个元素的存储位置之间都有固定的联系，因为可以从头结点查找任何一个元素（ X ）。（链表中的元素的存储位置，没有固定联系）

在单链表中，要取得某个元素，只要知道该元素的指针即可，因此，单链表是随机存取的存储结构（ X ）。（顺序表才是随机存储）

链表的每个结点中，都恰好包含一个指针（ X ）。（双向链表就有两个指针）

使用三元组表示稀疏矩阵的元素，有时并不能节省存储空间（ √ ）。（如果非零元个数太多，就不节省时间了）

由树转换成二叉树，其根结点的右子树总是空的（√ ）。（根结点没有兄弟）

后序遍历树和中序遍历与该树对应的二叉树，其结果不同（ X ）。（可能相同，也可能不同）

若有一个结点是某二叉树子树的中序遍历序列中的最后一个结点，则它必是该子树的前序遍历序列中的最后一个结点（√ ）。（都是最右的结点）

已知二叉树的前序遍历和后序遍历序列并不能唯一地确定这棵树，因为不知道树的根结点的左子树和右子树中分别包含哪些结点。（√ ）

有回路的图不能进行拓扑排序（ √ ）。

连通分量是无向图中的极小连通子图（ X ）。

散列法存储的基本思想是由关键码的值决定数据的存储地址（ √ ）。

散列表的查找效率取决于散列表造表时选取的散列函数和处理冲突的方法（ √ ）。

中序遍历二叉排序树的结点就可以得到排好序的结点序列（ √ ）。

在二叉排序树上插入新的结点时，不必移动其它结点，仅需改动某个结点的指针，由空变为非空即可（√）。

当待排序的元素很多时，为了交换元素的位置，移动元素要占用较多的时间，这是影响时间复杂性的主要因素（X）。（比较次数多才是主要因素）

对于 n 个记录的集合进行快速排序，所需要的平均时间是  $O(n \log_2 n)$ （√）。

对于 n 个记录的集合进行归并排序，所需要的平均时间是  $O(n \log_2 n)$ （√）。

堆中所有非终端结点的值均小于或等于（大于或等于）左右子树的值（√）。

顺序存储方式只能用于存储线性结构。（X）（完全二叉树也可以顺序存储）

对任何数据结构链式存储结构一定优于顺序存储结构。（X）

有向图的邻接矩阵是对称矩阵，无向图的邻接矩阵是非对称矩阵。（X）

所有二叉树的度均为 2。（X）（均小于等于 2）

满二叉树一定是完全二叉树，但完全二叉树不一定是满二叉树。（√）

循环链表不是线性表。（X）

线性表的特点是每个元素都有一个前驱和一个后继。（X）（首个元素没有前驱）

按中序遍历二叉排序树得到中序序列是一个递增有序序列。（√）

若哈夫曼树中含有 n 个结点，则整一颗哈夫曼树中必然有  $2n-1$  个结点。（√）

树可以转换成二叉树，但不是所有的二叉树都可以转换成树。（√）（若二叉树根节点有右孩子，则会转换为多棵树的森林，因此由树转换过来的二叉树的根节点没有右孩子。）

#### 四、简答题

线性表有两种存储结构：一是顺序表，二是链表。试问：

(1) 如果有 n 个线性表同时并存，并且在处理过程中各表的长度会动态发生变化，线性表的总数也会自动地改变。在此情况下，应选用哪种存储结构？为什么？

(2) 若线性表的总数基本稳定，且很少进行插入和删除，但要求以最快的速度存取线性表中的元素，那么，应采用哪种存储结构？为什么？

何谓队列的上溢现象？解决它有哪些方法，且分别简述其工作原理。

简述二叉树与度小于等于 2 的有序树之间的区别。

若有 5000 个元素，如果仅要求求出其中最大的 3 个元素，会采用哪种排序算法最节省时间。

有一个 2000 项的表，欲采用索引查找（分块）方法进行查找，问若每块长度为 20，平均查找长度是多少？

解析稠密图和稀疏图的定义，在邻接矩阵和邻接表两种图的存储结构中，稠密图适用哪种存储结构，稀疏图又适用哪种存储结构，为什么？

#### 五、应用题

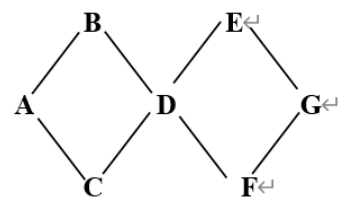
判断下列序列是否是堆。若不是堆，则把它们依次调整为堆。

(100, 85, 98, 77, 80, 60, 82, 40, 20, 10, 66);

已知一组元素为(46、25、78、62、18、34、12、40、73)，试画出按元素排列顺序输入而生成的一棵二叉排序树，并求其平均查找长度。

已知一棵二叉树先序遍历结果为 ABCDEFGHIJ，中序遍历的结果为 CBEDAHGIJF，试画出该二叉树。

设无向图 G 如下图：



试给出： 该图的邻接矩阵、该图的邻接表、该图的各项点的度；  
从 A 出发的“深度优先”遍历序列；从 A 出发的“广度优先”遍历序列

假定一个待散列存储的线性表为(32、75、63、48、94、25、36、18、70)，采用散列函数  $H(k) = k \text{ MOD } 11$ ，试画出以下两种情况的散列表，并求出在等概率情况下的平均查找长度。

假设用于通信的电文由字符集{a,b,c,d,e,f,g, h}中的字母构成。它们在电文中出现的频度分别为{27,10,12,6,30,3,21,15}。要求：

请为这 8 个字母设计哈夫曼编码，并画出对应的哈夫曼树；

计算该哈夫曼树的最小加权路径长度 WPL。

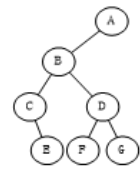
设图  $G = (V, E)$ ，其中  $V = \{A, B, C, D, E, F\}$ ，其邻接矩阵如下所示。若将该图用邻接表存储，请写出图的邻接表。按照 Prim 方法，从顶点 A 出发，求该网的最小生成树的产生过程，并计算该最小生成树的代价值。

$$A(G) = \begin{bmatrix} A & B & C & D & E & F \\ \begin{bmatrix} 0 & \infty & 10 & \infty & 30 & 100 \\ \infty & 0 & 5 & \infty & \infty & \infty \\ 10 & 5 & 0 & 50 & \infty & \infty \\ \infty & \infty & 50 & 0 & 20 & 10 \\ 30 & \infty & \infty & 20 & 0 & 60 \\ 100 & \infty & \infty & 10 & 60 & 0 \end{bmatrix} \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix}$$

请画出利用下面的顺序存储结构表示的二叉树，#表示相应的位置没有结点存在。

AB#CD###EFG####

写出该二叉树的先序遍历，中序遍历，后序遍历及层次遍历结果



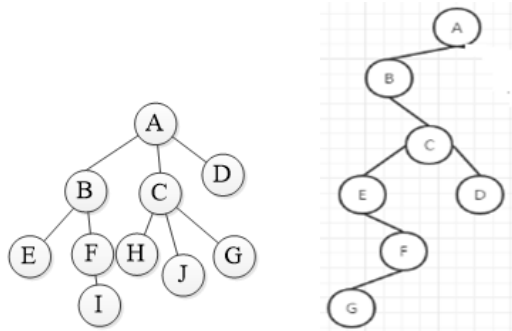
将此二叉树用仿真指针孩子双亲表示法存储。

	Parent	Lchild	Rchild
A	-1	1	-1
B	0	2	3
C	1	-1	4
D	1	5	6
E	2	-1	-1
F	3	-1	-1
G	3	-1	-1

对关键码序列(1, 5, 2, 4 , 6, 7, 8, 3)，执行升序排序，写出以下排序算法在每一趟排序结束时的关键码序列。

冒泡排序；简单选择排序 希尔排序 (d={5,2,1})

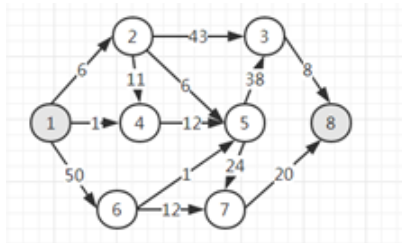
请把图 1 的树转换为二叉树，对二叉树进行前，中，后，层序遍历，把图 2 的二叉树转换为树，对树进行先根，后根，层序遍历。



有如下图所示的带权有向图 G，试回答以下问题。

给出 G 的一个拓扑序列

给出从顶点 1 到顶点 8 的最短路径



## 六、算法设计题

- 1、单链表的插入和删除，双向链表的插入和删除。
- 2、括号匹配算法。
- 3、二叉树的先序、中序、后续遍历的递归算法，求二叉树的高度的递归算法。
- 4、图的深度优先遍历的递归算法。