一．是非题（共 分，每题 分）

1. 数据结构可用三元式表示（D，S，P）。其中：D 是数据对象，S 是 D 上的关系，P 是对 D 的基本操作集。(f)
2 简单地说,数据结构是带有结构的数据元素的集合。(t)
3 判断带头结点的非空循环单链表（头指针为 L）中指针 p 所指结点是最后一个元素结点的条件是：p->next==L。(t)
4 线性表的链式存储结构具有可直接存取表中任一元素的优点。(f)
5 线性表的顺序存储结构优于链式存储结构。(f)
6. 在单链表 P 指针所指结点之后插入 S 结点的操作是：
P->next= S ; S-> next = P->next;。(f)
7 对于插入、删除而言，线性表的链式存储优于顺序存储。(t)
8. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。(f)
9. 栈和队列是操作上受限制的线性表。(t)
10. 队列是与线性表完全不同的一种数据结构。(f)
11. 队列是一种操作受限的线性表，凡对数据元素的操作仅限一端进行。(f)
12. 栈和队列也是线性表。如果需要，可对它们中的任一元素进行操作。(f)
13. 栈是限定仅在表头进行插入和表尾进行删除运算的线性表。(f)
14. 二叉树中每个结点有两个子结点，而对一般的树，则无此限制，所 以，二叉树是树的特殊情形。(f)
15 二叉树是一棵结点的度最大为二的树。(f)
16 赫夫曼树中结点个数一定是奇数。(t)
17 在二叉树的中序遍历序列中，任意一个结点均处在其左孩子结点的后面。(t)
18 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′的后序遍历 。(f)
19. 通常，二叉树的第 i 层上有 2^{i-1} 个结点。(f)
20. 中序线索二叉树的优点是便于在中序下查找直接前驱结点和直接后继结点。(t)
21 二叉树的先序遍历序列中，任意一个结点均处在其孩子结点的前面。(t)
22 由树结点的先根序列和后根序列可以唯一地确定一棵树。（t）
23 邻接多重表可以用以表示无向图，也可用以表示有向图。(f)
24 可从任意有向图中得到关于所有顶点的拓扑次序。(f)
25 有向图的十字链表是将邻接表和逆邻接表合二为一的链表表示形式。(t)
26 关键路径是 AOE 网中源点到汇点的最短路径。(f)
27 连通图 G 的生成树是一个包含 G 的所有 n 个顶点和 n-1 条边的子图。(f)
28 一个无向图的连通分量是其极大的连通子图。(t)
29 十字链表可以表示无向图，也可用以表示有向图。(f)
30 邻接表可以表示有向图，也可以表示无向图。（ t ）
31. 二叉排序树的平均查找长度为 O(log n)。(t)
32. 二叉排序树的最大查找长度与（LOG_2N）同阶。(f)
33 选用好的 HASH 函数可避免冲突。(f)
34 折半查找不适用于有序链表的查找。(t)
35. 对于目前所知的排序方法，快速排序具有最好的平均性能。(t)
36 对于任何待排序序列来说，快速排序均快于冒泡排序。(f)
37 在最坏情况下，堆排序的时间性能是 O(nlogn),比快速排序好(t)
38 快速排序具有最好的平均时间性能，它在任何时候的时间复杂度都是 O（n log n）。(f)
39. 字符串是数据对象特定的线性表。(t)

40. 空串与空格串是相同的。(f)
41. 对于一棵 m 阶的 B.树.树中每个结点至多有 m 个关键字.除根之外的所有非终端结点至少有 ┌m/2┐ 个关键字。(f)
42. 当二叉排序树是一棵平衡二叉树时，其平均查找长度为 O(log2n)。(t)
43. 广义表的表头和表尾都是广义表。(f)
44 二维数组是其数据元素为线性表的线性表。(t)

选择题。

1 从逻辑上可以把数据结构分成(c)。

A. 动态结构和静态结构　　B. 顺序组织和链接组织
C. 线性结构和非线性结构　　D. 基本类型和组合类型

2 线性表 L 在(b)情况下适于使用链表结构实现。

A. 不需修改 L 的结构　　B. 需不断对 L 进行删除、插入
C. 需经常修改 L 中结点值　　D. L 中含有大量结点

3 带头结点的单链表 L 为空的判断条件是 b 。
带头结点的循环链表 L 为空的判断条件是 _____ 。

A. L==null　　B. L->next==null
C. L->next==L　　D. L!=null

4 若顺序表中各结点的查找概率不等，则可用如下策略提高顺序查找的效率：若找到指定的结点，将该结点与其后继（若存在）结点交换位置，使得经常被查找的结点逐渐移至表尾。以下为据此策略编写的算法，请选择适当的内容，完成此功能。
顺序表的存储结构为：

```
typedef struct{
    ElemType *elem; //数据元素存储空间，0 号单元作监视哨
    int        length; //表长度
  }SSTable;
int search_seq(SSTable ST,KeyType key)
{ //在顺序表 ST 中顺序查找关键字等于 key 的数据元素。
//若找到，则将该元素与其后继交换位置，并返回其在表中的位置，否则为 0。
ST.elem[0].key=key;
i=ST.length;
while(ST.elem[i].key!=key)   f   ;
if(    G    )
  {ST.elem[i]←→ST.elem[i+1];
    _____e_____;
  }
return i;
}
```

A. i>0　　B. i>=0　　C. i<ST.length　　D. i<=ST.length
E. i++　　F. i--　　G. A 和 C 同时满足　　H. B 和 D 同时满足

5 若入栈顺序为 A、B、C、D、E，则下列(d)出栈序列是不可能的。

A．A、B、C、D、E　　B．B、C、D、A、E
C．C、D、B、E、A　　D．D、E、C、A、B

6 递归程序可借助于(　c　)转化为非递归程序。

a.线性表　　b.队列　　c: 栈　　d.数组

7 在下列数据结构中(　c　)具有先进先出（FIFO）特性，

(　b　)具有先进后出(FILO）特性。

a．线性表　　b．栈　　c．队列　　d．广义表

8 若对编号为 1，2，3 的列车车厢依次通过扳道栈进行调度，不能得到（　e）的序列。

a:1,2,3　　b:1,3,2　　c:2,1,3　　d:2,3,1　　e:3,1,2　　f:3,2,1

9 在计算递归函数时，如不用递归过程，应借助于(b)　这种数据结构。

A. 线性表　　B. 栈　　C. 队列　　D. 双向队列

10　若带头结点的链表只设尾结点指针。下列选择中（　c　）最适用于队列。

A）单链表　　B）双向链表　　C 循环单链表　　D）双向循环链表

11 栈和队列的一个共同点是(c　)。

A. 都是先进先出　　B. 都是先进后出

C. 只允许在端点处插入和删除元素　　D. 没有共同点

12 循环队列用数组 A[0..m-1]存放其元素值，设头尾指针分别为 front 和 rear，则当前队列中的元素个数是(　c　)。

A. rear-front-1　　B. Rear-front+1

C. (rear-front+m)%m　　D. Rear-front

13 如下关于串的陈述中，正确的是(a,　c　)。

A. 串是数据元素类型特殊的线性表　　B. 串中的元素是字母

C. 串中若干个元素构成的子序列称为子串　　D. 空串即为空格串

14 对字符串 s='data-structure' 执行操作 replace(s,substring(s,6,8),'bas')

的结果是（　b　）。

a: 'database'　b: 'data-base'　c: 'bas'　d: 'data-basucture'

15 设有二维数组 $A_{5\times7}$,每一元素用相邻的 4 个字节存储，存储器按字节编址.

已知 A 的起始地址为 100。则按行存储时，元素 A_{06} 的第一个字节的地址是（d　）

按列存储时，元素 A_{06} 的第一个字节的地址是（ a　）

a: 220　　b:　200　　c: 140　　d:　124

16 对广义表 A=((a,(b)),(c,()),d）执行操作 gettail(gethead(gettail(A)))

的结果是：（　b　）。

a:()　　b: (())　　c: d　　d: (d)

17 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 7, 19, 22, 6, 32, 14。 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 7 的字符编码是（　g　），频率为 32 的字符编码是（　c　）。

a: 00　　b: 01　　c: 10　　d: 11

e: 011　　f: 110　　g: 1110　　h:1111

18 对二叉排序树（　c　）可得到有序序列。

a:按层遍历　　b:前序遍历　　c:中序遍历　　d:后序遍历

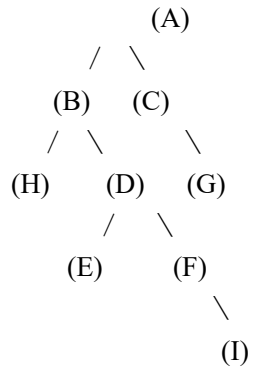
19 设一棵二叉树 BT 的存储结构如下:

	1	2	3	4	5	6	7	8
lchild	2	3	0	0	6	0	0	0
data	A	B	C	D	E	F	G	H
rchild	0	5	4	0	8	7	0	0

其中 lchild，rchild 分别为结点的左、右孩子指针域，data 为结点的数据域。则该二叉树的高度为(d)；
第 3 层有(a)个结点（根结点为第 1 层）。

A．2 B. 3 C. 4 D. 5

20 先序遍历图示二叉树可得到（ a ）的序列。



a) A B H D E F I C G
b) H B E D F I A C G
c) H E I F D B G C A

21 在有 n 个结点的二叉树的二叉链表表示中，空指针数 （ b ）。

a.不定 b.n+1 c.n d.n-1

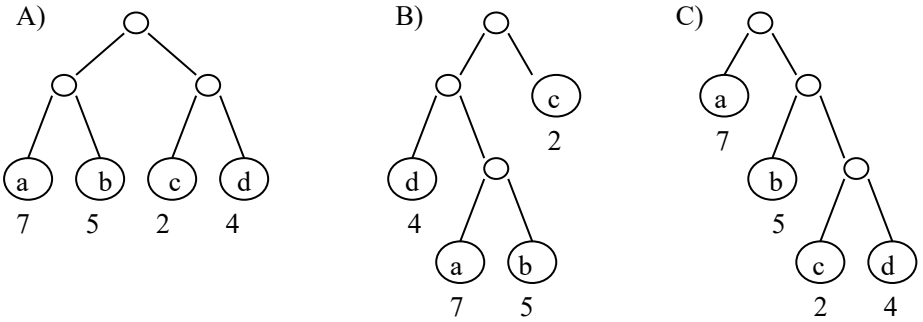
22 若某二叉树有 20 个叶子结点，有 20 个结点仅有一个孩子，则该二叉树的总结点数是(c)。

A．40 B. 55 C. 59 D. 61

23 已知某二叉树的先序遍历次序为 abcdefg 中序遍历次序为 badcgfe，
则该二叉树的后序遍历次序为（ c ）。层次遍历次序为（ a ）。

a: abcdefg b: cdebgfa c: bdgfeca d: edcgfba

.24 图示的三棵二叉树中(c)为最优二叉树。



25 已知某二叉树的后序遍历和中序遍历次序分别为 DBFGECA 和 BDACFEG。
则其先序遍历次序为（ b ），层次遍历次序为（ a ）。

a: abcdefg b: abdcefg c: abcdfeg d: abcdegf

26 已知某树的先根遍历次序为 abcdefg 后根遍历次序为 cdebgfa。
若将该树转换为二叉树,其后序遍历次序为（ d ）。

a: abcdefg b: cdebgfa c: cdegbfa d: edcgfba

27 设 x 和 y 是二叉树中的任意两个结点，若在先根序列中 x 在 y 之前，而在后根序列中 x 在 y 之后，则 x 和 y 的关系是(　c　)。

A. x 是 y 的左兄弟　　B. x 是 y 的右兄弟

C. x 是 y 的祖先　　D. x 是 y 的子孙

28 用三叉链表作二叉树的存储结构，当二叉树中有 n 个结点时，有(　d　)个空指针。

A. n-1　　B. n　　C. n+1　　D. n+2

29 对一棵完全二叉树进行层序编号。则编号为 n 的结点若存在右孩子,其位序是(　d)。编号为 n 的结点若存在双亲,其位置是(a　)。

a: n/2　　b: 2n　　c:2n-1　　d:2n+1　　e:n　　f: 2(n+1)

30 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 对应的二叉树根结点的右子树上的结点个数是(d)。

A. m1　　B. m1+m2　　C. m3　　D. m2+m3

31 下列二叉树中，(a　)可用于实现符号不等长高效编码。

a:最优二叉树　　b:次优查找树　　c:二叉平衡树　　d:二叉排序树

32 邻接表存储结构下图的深度优先遍历算法类似于二叉树的(　a　)遍历。

A. 先根　　B. 中根　　C. 后根　　D. 层次

33 设无向图 G = (V,E)和 G'= (V',E')，若 G'是 G 的生成树，则下面不正确的说法是(　b　)。

A. G'是 G 的子图　　B. G'是 G 的连通分量

C. G'是 G 的无环子图　　D. G'是 G 的极小连通子图且 V'= V

34 任何一个连通图的最小生成树(b)。

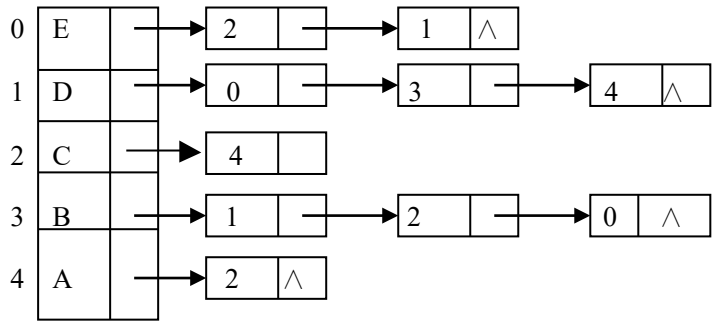
A．只有一棵　　B. 有一棵或多棵　　C. 一定有多棵　　D. 可能不存在

e　　f　　e　　f

35 深度优先遍历图使用了数据结构（b　），而广度优先遍历图使用了数据结构（ c ）。

A）数组　　B）栈　　C）队列　　D）线性表

36 已知某有向图的邻接表存储结构如图所示。



根据存储结构依教材中的算法其深度优先遍历次序为（　d　　）。

广度优先遍历此序为（　c　　）。各强连通分量的顶点集为（　　h　　）。

a:　abcde.　　b:　edcba.　　c:　ecdab.　　d:　ecadb.

e:　abc 及 ed　　f: bc 及 aed　　g:　ab 及 ced　　h:　ac 及 bed

37 下列查找方法中（ a ）适用于查找单链表。

A）顺序查找　　B）折半查找　　C）分块查找　　D）hash 查找

38 下列算法中（c　）适用于求图的最小代价生成树。（ b　）能对图作广度优先遍历。

A）DFS 算法　　B）BFS 算法　　C）Prim 算法　　D）Dijkstra 算法

39 关键路径是指在只有一个源点和一个汇点的有向无环网中源点至汇点（　c　）的路径。

a:弧的数目最多　　b:弧的数目最少　　c:权值之和最大　　d:权值之和最小

40 希表的查找效率取决于（　d　　）。

a: 哈希函数　　b:处理冲突的方法。　　c:哈希表的装填因子。　　d:以上都是

41　在 Hash 函数 H(k)=k MOD m 中，一般来说，m 应取(　　c)。

A. 奇数　　B. 偶数　　C. 素数　　D. 充分大的数

42 在顺序表查找中，为避免查找过程中每一步都检测整个表是否查找完毕，可采用__a__方法。

A.设置监视哨　　B.链表存贮　C.二分查找　　D.快速查找

43 静态查找表和动态查找表的区别在于(　b　)。

A. 前者是顺序存储，而后者是链式存储

B. 前者只能进行查找操作，而后者可进行查找、插入和删除操作

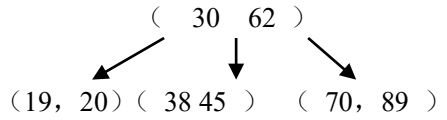
C. 前者只能顺序查找，而后者只能折半查找

D. 前者可被排序，而后者不能被排序

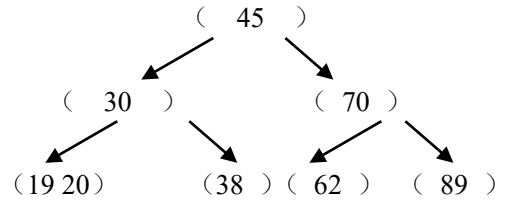
44 在一个含有 n 个元素的有序表上进行折半查找，找到一个元素最多要进行(　b　)次元素比较。

A．$\lfloor \log_2(n) \rfloor$　　B.$\lfloor \log_2(n) \rfloor+1$　　C.$\lfloor \log_2(n+1) \rfloor$　　D.$\lfloor \log_2(n+1) \rfloor+1$

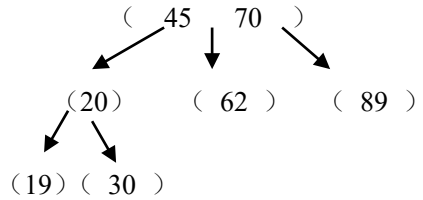
45 设输入序列为 20, 45, 30, 89, 70, 38, 62，19 依次插入到一棵 2-3 树中(初始状态为空)，该 B-树为（　b　）。再删除 38，该 B-树为（　f　）。



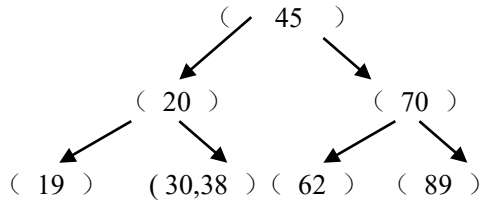
a:



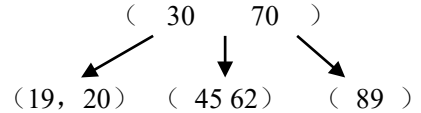
b:



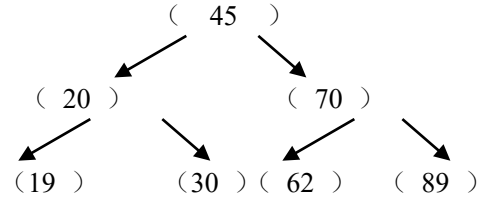
c:



d:

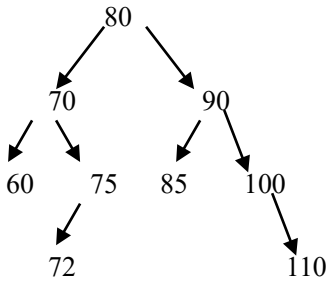


e:

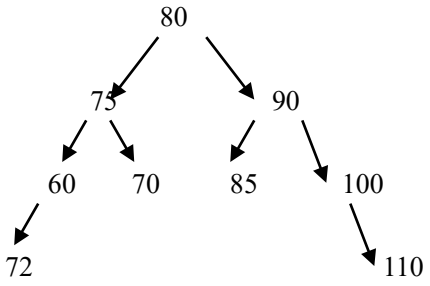


f:

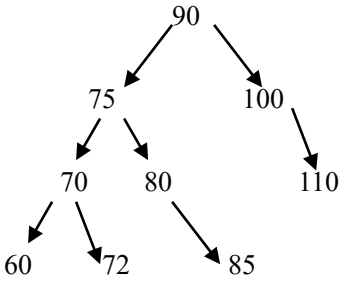
46 根据插入次序（80，90，100，110，85，70，75，60，72）建立二叉排序树。
图（ a ）是最终变化的结果。若仍以该插入次序建立平衡二叉树。图（ c ）是最终变化的结果。



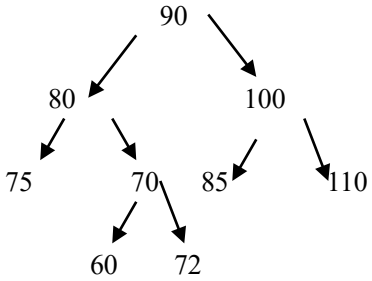
a:



b:



c:



d:

47 若有序表中关键字序列为：14，20，25，32，34，45，57，69，77，83，92。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(c)。查找 32 时需进行(c)次比较。

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

48 已知哈希表地址空间为 A[9]，哈希函数为 H(k)=k mod 7，采用线性探测再散列处理冲突。若依次将数据序列：76,45,88,21,94,77,17 存入该散列表中，则元素 17 存储的下标为(h)；在等概率情况下查找成功的平均查找长度为(c)。

A. 0 B. 1 C. 2 D. 3
E. 4 F. 5 G. 6 H. 7

49 若从二叉树的根结点到其它任一结点的路径上所经过的结点序列按其关键字递增有序，则该二叉树是(c)。

A. 二叉排序树 B. 赫夫曼树 C. 堆 D. 平衡二叉树

50 当待排序序列的关键字次序为倒序时，若需为之进行正序排序，下列方案中(d)为佳。

A. 起泡排序 B. 快速排序
C. 直接插入排序 D. 简单选择排序

51 下列排序算法中，(d)算法可能会出现：初始数据有序时，花费的时间反而最多。

A. 堆排序 B. 起泡排序 C. 归并排序 D. 快速排序

52 在下列排序方法中，（ c ）方法平均时间复杂度为 0(nlogn)，
最坏情况下时间复杂度为 0(n2)；（ d ）方法所有情况下时间复杂度均为 0(nlogn)。

a. 插入排序 b. 希尔排序 c. 快速排序 d. 堆排序

53 已知一组待排序的记录关键字初始排列如下：56,26,86,35,75,19,77,58,48,42
下列选择中（ d ）是快速排序一趟排序的结果。（ c ）是希尔排序
（初始步长为 3）一趟排序的结果。（ a ）是初始堆（大堆顶）。

A）86,75,77,58,42,19,56,35,48,26.
B）26,56,35,75,19,77,58,48,42,86.
C）35,26,19,42,58,48,56,75,86,77.
D）42,26,48,35,19,56,77,58,75,86.

三．填空题

1 数据结构通常有下列 4 类基本结构：集合、_____ 、树型结构、图型结构。

2 设单链表中结点形式为 | data | next | ，若单链表长度大于等于 2，指针 p 指向表中某个结点且 p->next 非空，此时若要删除指针 p 所指的结点，可以通过如下方法进行：将 p 所指结点的后继的元素值复制到该结点，然后删除其后继结点。相应的语句序列为：

_____; _____; _____ ; _____;

3 线性表的顺序存储结构是以_____来表示数据元素之间的逻辑关系的。

4 已知 P 是单链表中某一结点的指针，P 既不是首元结点也不是尾元结点，Q 是 P 的 前驱结点指针。当删除 P 结点时，链表的链接可用语句（ ）实现。

5 已知某树的先根遍历次序为 abcdefg 后根遍历次序为 cdebgfa。
若将该树转换为二叉树,其后序遍历次序为（ ）。层次遍历次序为（ ）。

6 已知某二叉树的先序遍历次序为 afbcdeg 后序遍历次序为 cedbgfa。
其后序遍历次序为（ ）。层次遍历次序为（ ）。

7 在二叉树的第 i 层上至少有_____个结点, 至多有_____个结点 ,
深度为 k 的二叉树至多有_____个结点.

8 对树的遍历有先序遍历树和后序遍历树。若以二叉链表作树的存储结构,
则树的先序遍历可借用二叉树的_____遍历算法来实现,
而树的后序遍历可借用二叉树的_____遍历算法来实现。

9 设高度为 h 的二叉树上只有度为 0 和度为 2 的结点，则此类二叉树中所包含的结点数至少是_____，至多是_____。

10 对任何一棵二叉树 T,若其终端结点数为 n0.度为 2 的结点为 n2,则 n0 与 n2 的关系为 ( )。

11 如果对完全二叉树中结点从 1 开始按层进行编号，设最大编号为 n；那么，可以断定编号为 i (i>1)的结点的父结点编号为( )；所有编号（ ）的结点为叶子结点。

12 n 个顶点的连通图至少有_____条边，至多有_____条边。

13 对于图的存储结构有（ ）、（ ）( ) ( )等方法。

14 在一个无向图的邻接表中，若表结点的个数是 m，则图中边的条数是_____条。

15 若有序表中关键字序列为：12，22，33，44，55，66，77，88，99 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 99 时需进行（ ）次比较。

16 在哈希表中，处理冲突的方法有开放定址法，_____， _____等。

17 在二叉树的第 i 层上至少有_____个结点, 至多有_____个结点 ,深度为 k 的二叉树至多有_____个结点.

18 对于一棵高度为 K 的二叉排序树，结点数最少可有_____个，最多可有_____个。

19 用_____遍历对二叉排序树进行访问可得到有序序列。

20 已知 Hash 函数为 H（K）=K mod 13 ，散列地址为 0 --14，用二次探测再散列处理冲突，关键字（23，34，56，24，75，12，49, 52，36，92）的分布如图，则平均成功的查找长度为（　　）、平均失败的查找长度为（　　）。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
52	36	92		56				34		23	24	75	12	49

21 一棵m阶的B-树，第一层至少有一个结点；第二层至少有 2 个结点，除根之外的所有非终端结点至少有（　　）棵子树，树中每个结点至多有（　　）棵子树。

22 在哈希表中，处理冲突的方法有开放定址法，_____ ，_____ ，_____ 。

23 哈希表的查找效率取决于（　　）(　　)和（　　）。

24 高度为 4 (包含不带关键字的叶子结点层)的 7□阶□B□树最少有_____个关键字，最多有_____个关键字；如果其中的某结点正好有 2 个儿子，那么，该结点必定是_____结点。

25 对 n 个元素的序列进行内部排序，若用起泡排序法，最少的比较次数是_____，最多的比较次数是_____。

25 (算法填空)

```
Status Preordertraverse(Bitree T,Status(*Visit)(Telemtype e)) {
//先序非递归遍历二叉树。
Initstack ( S );     Push ( S,T );
While ( !stackempty( S ) )
        {   While ( gettop( S, p )&& _____ )
                {   if (!Visit (p->data ) ) return ERROR;
                    _____ ;
                }
            Pop ( S , p );
            if   (   _____   )
 {   _____;     push( S, p->rchild );   }
}
return ok;
}
```

26 (算法填空)

下列算法试图完成在数组 A 中搜索有无关键字 key，若有，返回数组下标,若无，返回-1。在“_____”处填上合适的内容,完成该算法。

```
int   BinarySearch (keytype A [], int low,int high, keytype key )
{
while     (_____)
          middle = (low+high) /2;
if (_____)
                  return middle;
```

```
        if    (key < A[middle])
              _____;
        else
_____;
}
return -1;
} //end of BinarySearch
```

27 (算法填空)

下列函数为堆排序中的堆调整过程（调整 H.r[s]的关键字，使 H.r[s..m]成为一小顶堆）。请在“_____”处填上合适的内容,完成该算法。

```
Void heapadjust( heaptype @ H , int s , int m ) {
rc=H.r[s];
for (j=2*s;j<=m;j*=2) {
if (j<m &&  _____  ) ++j;
if (  _____  ) break;
H.r[s]=H.r[j]; s=j;
}
_____ ;
}//heapadjust
```

图示结构题

1 已知在电文中只出现频率为 (5,26,7,23,20,19)的 6 个字符，
画出你建的哈夫曼树，并给出其哈夫曼编码。

2.已知某二叉树的后序遍历和中序遍历次序分别为 DBFGECA 和 BDACFEG
请画出该二叉树，并为之建立先序线索。

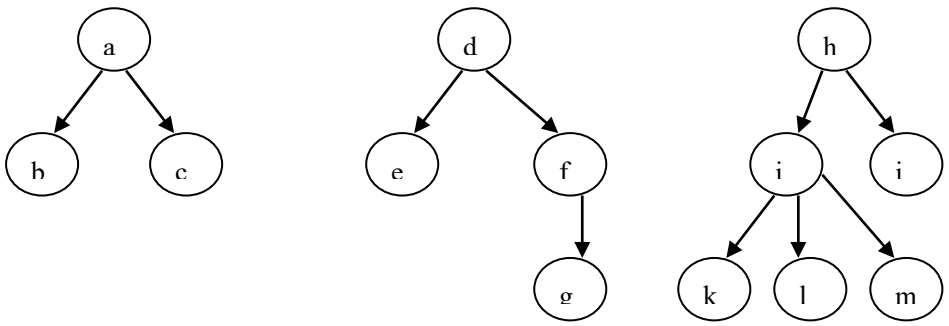
3 已知某二叉树的先序遍历次序为：a,b,c,d,e,f,g.中序遍历次序为：b,a,d,f,e,g,c
画出该二叉树，并在该二叉树上建立中序线索。

4 某二叉树的中序遍历次序为 BEGFDAC， 先序遍历次序为 ABDEFGC。
试画出该二叉树，并为之建立中序线索（图示之）。

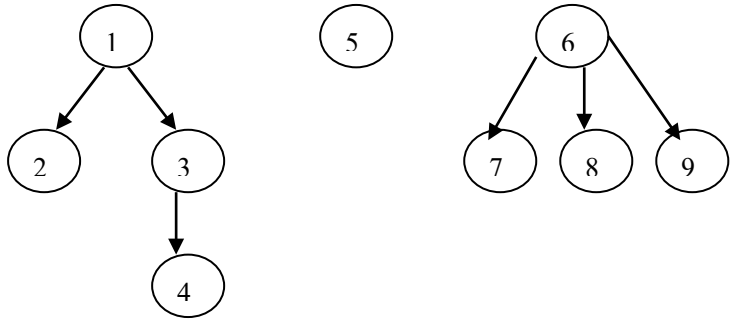
5 已知某二叉树的后序遍历和中序遍历次序分别为 FBEDGCA 和 FBADECG，
请构造并画出该二叉树。

6 设某一电文只出现 a,b,c,d,e,f,g 7 个字母；出现频率分别为 30%,10%,05%,04%,13%,18%
及 20%，请给出各字母的哈夫曼编码。

7 将图示森林转换为二叉树，并对该二叉树先序全序线索化。



8 将图示森林转换为二叉树，并对该二叉树中序全序线索化。



9 某二叉树的结点数据采用顺序存储表示如下：

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	B	C		D		E			F				G	H					I

（1）试画出此二叉树的图形表示。

（2）将此二叉树看作森林的二叉树表示，试将它还原为森林。

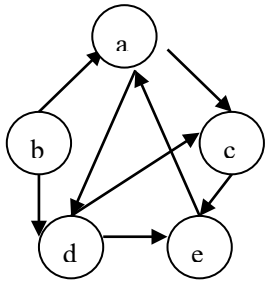
10 已知某有向图如图所示：

1）给出其十字链表存储结构

2）给出其深度优先遍历次序。

3）给出其广度优先遍历次序。

4）给出各强连通分量。



11 设输入序列为 20,45,30,89,70,38,62,19，依次插入到一棵 2-3 树中(初始状态为空)，请画出该 B-树。

12 右图为一棵 3 阶 B－树。

1)画出在该树上插入元素 15 后的 B－树。

2)接着，再删除元素 35，画出删除后的 B－树。

（20，25）

/　　|　　\

(10，14)（21）（35）

13 已知 Hash 函数为 H（K）=K mod 13 ，散列地址为 0 --14，用线性探测再散列处理冲突，给出关键字（56，34，68，23，16，70，48，35，83，12，14，57）在散列地址的分布。并指出平均成功的查找长度是多少?

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

14 根据插入次序（20，30，70，60，10，100，110，90，80。）建立平衡的二叉排序树。

15 设哈希表长为 16，哈希函数为 H(key)=key mod 13，用开放定址法的二次探测再散列处理冲突（$d_i=1^2$，-1^2，2^2，-2^2，3^2，-3^2……）。依次存入 12 个元素：56，82，17,24, 36,21,83,96,13,34,57,50。请画出它们在表中的分布情形。

16 已知待排序序列为：25,12,9,20,7,31,24,35,17,10，试写出：
(1). 堆排序初始建堆(大顶堆)的结果；
(2). 以第一个元素为枢轴的快速排序一趟扫描的结果；
(3). 希尔排序第一趟(增量为 5)的结果。

算法设计题

1 设有一个带头结点、元素按值递增有序的单链表，结点的类型定义如下：

```
typedef struct LNode
{ int data;
  struct LNode *next;
} LNode, *LinkList;
```

编写算法，删除其中所有值相同的多余元素结点

2 某线性表中元素以降序排列，现要插入一个元素 X，插入后该线性元素仍保持降序。线性表采用带头结点单链表方式存贮。□请编写该插入算法。

3 编写在一有序顺序表中插入数据元素 X 的算法 INSERT（L， X）。

4 写一算法，Delete(linklist &L，X) ,删除单链表中所有值为 X 的结点。
单链表结点的类型定义如下：

```
typedef struct LNode {
        int data;
        struct LNode *next;
} LNode, *Linklist;
```

5 写一算法，Contrary(linklist &L) ,对一带头结点且仅设尾指针 L 的循环单链表就地逆置。（即表头变表尾，表尾变表头。）

6 已知线性表中的元素以值递增有序排列，并以带头结点的单链表作存储结构。试写一高效的算法，删除表中所有值大于 mink 且小于 maxk 的元素（若表中存在这样的元素）同时释放被删结点空间，并分析你的算法的时间复杂度。
单链表结点的类型定义如下：

```
typedef struct LNode {
        int data;
        struct LNode *next;
} LNode, *Linklist;
```

7 写一算法，将带头结点的有序单链表 A 和 B 合并成一新的有序表 C。(注:不破坏 A 和 B 的原有结构.)Merge(Linklist A, Linklist B, Linklist &C)

8 写一算法 Oplinklist(linklist L,int i;int j)
删除单链表中第 i 个元素,并将之插入至原表中的第 j 个元素之前.

9 写出求单链表长度算法 int length(linklist L)

10 若将循环队列 Q 的结构定义为：

```
#define   m   100      //最大队列长度
typedef struct
    {   QElemType   *base;      //存储空间基址
```

```
    int rear;      //尾指针，若队列不空，指向队尾元素
    int length;    //当前队列的长度，即元素个数
} SqQueue;
```

试写出相应初始化、入队列和出队列的三个函数。

11 二叉树用二叉链表存储表示。

```
typedef struct BiTNode {
    TelemType   data;
    Struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

试编写销毁二叉树 T 的算法 DestroyBiTree (BiTree &T)。

12 二叉树用二叉链表存储表示。

```
typedef struct BiTNode {
    TelemType   data;
    Struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

试编写算法，求元素值为 x 的结点的左孩子（返回 x 的左孩子的指针）。

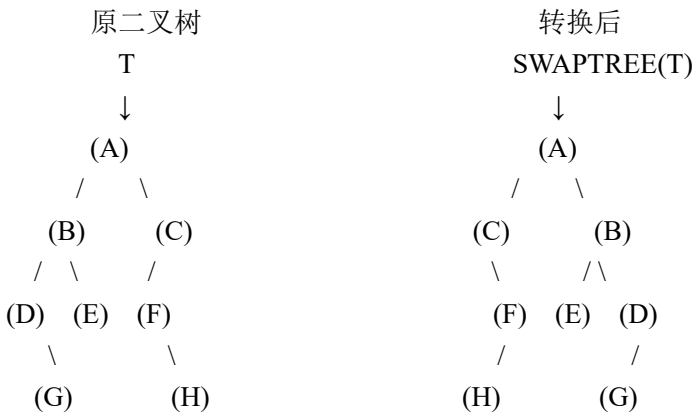
13 设计一算法，计算给定二叉树 T 中度为 2 的结点个数。

14 编一算法：按层序遍历二叉树 T。

15 试编写先序遍历二叉树 T 的递归算法 PreorderBiTree (BiTree &T)。

16 写出一个将树中每个结点的左右孩子对换的算法

SWAPTREE(T)　　即如:



17 二叉树用二叉链表存储表示。

试编写后序遍历二叉树 T 的递归算法 PostorderBiTree (BiTree T)。

18 写一个计算二叉树中叶子结点个数的递归算法。