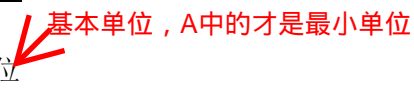


《数据结构与算法》复习题

一、选择题。

1. 在数据结构中，从逻辑上可以把数据结构分为__C__。
A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
C. 线性结构和非线性结构 D. 内部结构和外部结构
2. 数据结构在计算机内存中的表示是指__A__。
A. 数据的存储结构 B. 数据结构 C. 数据的逻辑结构 D. 数据元素之间的关系
3. 在数据结构中，与所使用的计算机无关的是数据的__A__结构。
A. 逻辑 B. 存储 C. 逻辑和存储 D. 物理
4. 在存储数据时，通常不仅要存储各数据元素的值，而且还要存储__C__。
A. 数据的处理方法 B. 数据元素的类型
C. 数据元素之间的关系 D. 数据的存储方法
5. 在决定选取何种存储结构时，一般不考虑 __A__。
A. 各结点的值如何 B. 结点个数的多少
C. 对数据有哪些运算 D. 所用的编程语言实现这种结构是否方便。
6. 以下说法正确的是__D__。
A. 数据项是数据的基本单位 
B. 数据元素是数据的最小单位
C. 数据结构是带结构的数据项的集合
D. 一些表面上很不不同的数据可以有相同的逻辑结构
7. 算法分析的目的是__C__，算法分析的两个主要方面是__A__。
(1) A. 找出数据结构的合理性 B. 研究算法中的输入和输出的关系
C. 分析算法的效率以求改进 C. 分析算法的易读性和文档性
(2) A. 空间复杂度和时间复杂度 B. 正确性和简明性
C. 可读性和文档性 D. 数据复杂性和程序复杂性
8. 下面程序段的时间复杂度是__ $O(n^2)$ __。

```
s=0;
for( I=0; i<n; i++)
    for(j=0;j<n;j++)
        s+=B[i][j];
sum = s ;
```
9. 下面程序段的时间复杂度是__ $O(n*m)$ __。

```
for( i=0; i<n; i++)
    for(j=0;j<m;j++)
        A[i][j] = 0;
```

10. 下面程序段的时间复杂度是 $O(\log_3 n)$ 。
- ```
i = 0;
while (i <= n)
 i = i * 3;
```
11. 在以下的叙述中，正确的是 B。
- A. 线性表的顺序存储结构优于链表存储结构
  - B. 二维数组是其数据元素为线性表的线性表
  - C. 栈的操作方式是先进先出 **先进后出**
  - D. 队列的操作方式是先进后出 **先进先出**
12. 通常要求同一逻辑结构中的所有数据元素具有相同的特性，这意味着 B。
- A. 数据元素具有同一特点
  - B. 不仅数据元素所包含的数据项的个数要相同，而且对应的数据项的类型要一致
  - C. 每个数据元素都一样
  - D. 数据元素所包含的数据项的个数要相等
13. 链表不具备的特点是 A。
- A. 可随机访问任一结点 **不足之处**
  - B. 插入删除不需要移动元素
  - C. 不必事先估计存储空间
  - D. 所需空间与其长度成正比
14. 不带头结点的单链表 head 为空的判定条件是 A。
- A. head == NULL
  - B. head->next == NULL
  - C. head->next == head
  - D. head != NULL
15. 带头结点的单链表 head 为空的判定条件是 B。
- A. head == NULL
  - B. head->next == NULL
  - C. head->next == head
  - D. head != NULL
16. 若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点，则采用 D 存储方式最节省运算时间。
- A. 单链表
  - B. 给出表头指针的单循环链表
  - C. 双链表
  - D. 带头结点的双循环链表
17. 需要分配较大空间，插入和删除不需要移动元素的线性表，其存储结构是 B。
- A. 单链表
  - B. 静态链表
  - C. 线性链表
  - D. 顺序存储结构
18. 非空的循环单链表 head 的尾结点（由 p 所指向）满足 C。
- A. p->next == NULL
  - B. p == NULL
  - C. p->next == head
  - D. p == head
19. 在循环双链表的 p 所指的结点之前插入 s 所指结点的操作是 D。
- A. p->prior = s; s->next = p; p->prior->next = s; s->prior = p->prior
  - B. p->prior = s; p->prior->next = s; s->next = p; s->prior = p->prior

C.  $s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior; p \rightarrow prior = s; p \rightarrow prior \rightarrow next = s$

D.  $s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior; p \rightarrow prior \rightarrow next = s; p \rightarrow prior = s$

20. 如果最常用的操作是取第  $i$  个结点及其前驱, 则采用 D 存储方式最节省时间。

A. 单链表    B. 双链表    C. 单循环链表    D. 顺序表    可随机访问

21. 在一个具有  $n$  个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是 B。

for(  $p = head \rightarrow next; p \rightarrow data < data; p = p \rightarrow next$  );

A.  $O(1)$     B.  $O(n)$     C.  $O(n^2)$     D.  $O(n \log 2n)$

22. 在一个长度为  $n$  ( $n > 1$ ) 的单链表上, 设有头和尾两个指针, 执行 B 操作与链表的长度有关。

A. 删除单链表中的第一个元素  
B. 删除单链表中的最后一个元素    需要遍历至最后  
C. 在单链表第一个元素前插入一个新元素  
D. 在单链表最后一个元素后插入一个新元素

23. 与单链表相比, 双链表的优点之一是 D。

A. 插入、删除操作更简单  
B. 可以进行随机访问  
C. 可以省略表头指针或表尾指针  
D. 顺序访问相邻结点更灵活    一般的话, 要访问上一节点的又从 head 起。

24. 如果对线性表的操作只有两种, 即删除第一个元素, 在最后一个元素的后面插入新元素, 则最好使用 B。

A. 只有表头指针没有表尾指针的循环单链表  
B. 只有表尾指针没有表头指针的循环单链表    表头的话, 要在最后插入元素的话就得遍历至最后!  
C. 非循环双链表  
D. 循环双链表

25. 在长度为  $n$  的顺序表的第  $i$  个位置上插入一个元素 ( $1 \leq i \leq n+1$ ), 元素的移动次数为:

A。     $i$  之后的  $(n-i+1)$  个元素都要移动!

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

26. 对于只在表的首、尾两端进行插入操作的线性表, 宜采用的存储结构为 C。

A. 顺序表    B. 用头指针表示的循环单链表  
C. 用尾指针表示的循环单链表    D. 单链表

27. 下述哪一条是顺序存储结构的优点? C。

A 插入运算方便    B 可方便地用于各种逻辑结构的存储表示  
C 存储密度大    D 删除运算方便    A, D 为不足, 此时需要移动大量元素!

28. 下面关于线性表的叙述中, 错误的是哪一个? B。

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。    便于随机访问!  
C 线性表采用链式存储, 不必占用一片连续的存储单元  
D 线性表采用链式存储, 便于进行插入和删除操作。

- C 线性表采用链式存储，不必占用一片连续的存储单元  
D 线性表采用链式存储，便于进行插入和删除操作。

29. 线性表是具有  $n$  个 B 的有限序列。  
A. 字符    B. 数据元素    C. 数据项    D. 表元素
30. 在  $n$  个结点的线性表的数组实现中，算法的时间复杂度是  $O(1)$  的操作是 A。  
A. 访问第  $i$  ( $1 \leq i \leq n$ ) 个结点和求第  $i$  个结点的直接前驱 ( $1 < i \leq n$ )  
B. 在第  $i$  ( $1 \leq i \leq n$ ) 个结点后插入一个新结点  
C. 删除第  $i$  ( $1 \leq i \leq n$ ) 个结点  
D. 以上都不对
31. 若长度为  $n$  的线性表采用顺序存储结构，在其第  $i$  个位置插入一个新元素的算法的时间复杂度为 C。  
A.  $O(0)$     B.  $O(1)$     C.  $O(n)$     D.  $O(n^2)$
32. 对于顺序存储的线性表，访问结点和增加、删除结点的时间复杂度为 C。  
A.  $O(n) O(n)$     B.  $O(n) O(1)$     C.  $O(1) O(n)$     D.  $O(1) O(1)$
33. 线性表  $(a_1, a_2, \dots, a_n)$  以链式方式存储，访问第  $i$  位置元素的时间复杂度为 C。  
A.  $O(0)$     B.  $O(1)$     C.  $O(n)$     D.  $O(n^2)$
34. 单链表中，增加一个头结点的目的是为了 C。  
A. 使单链表至少有一个结点    B. 标识表结点中首结点的位置  
C. 方便运算的实现    D. 说明单链表是线性表的链式存储  
使对单链表的插入、删除操作不需要区分是否为空表或是否在第一个位置进行，从而与其他位置的插入、删除操作一致
35. 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 B。  
A.  $p \rightarrow next = s; s \rightarrow next = p \rightarrow next$     B.  $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$   
C.  $p \rightarrow next = s; p \rightarrow next = s \rightarrow next$     D.  $p \rightarrow next = s \rightarrow next; p \rightarrow next = s$
36. 线性表的顺序存储结构是一种 A。  
A. 随机存取的存储结构    B. 顺序存取的存储结构  
C. 索引存取的存储结构    D. Hash 存取的存储结构
37. 栈的特点是 B，队列的特点是 A。  
A. 先进先出    B. 先进后出
38. 栈和队列的共同点是 C。  
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
39. 一个栈的进栈序列是  $a, b, c, d, e$ ，则栈的不可能的输出序列是 C。  
A.  $edcba$     B.  $decba$     C.  $dceab$     D.  $abcde$   
 $i \dots j \dots k$  进栈不可能出现  $k \dots i \dots j$  出栈！

40. 设有一个栈，元素依次进栈的顺序为 A、B、C、D、E。下列\_\_C\_\_是不可能的出栈序列。

- A. A,B,C,D,E    B. B,C,D,E,A    C. E,A,B,C,D    D. E,D,C,B,A

41. 以下\_\_B\_\_不是队列的基本运算？

- A. 从队尾插入一个新元素    B. 从队列中删除第 i 个元素  
C. 判断一个队列是否为空    D. 读取队头元素的值

42. 若已知一个栈的进栈序列是 1, 2, 3, ..., n，其输出序列为 p<sub>1</sub>, p<sub>2</sub>, p<sub>3</sub>, ..., p<sub>n</sub>，若 p<sub>1</sub> = n，则 p<sub>i</sub> 为\_\_C\_\_。

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

说明是全部数据  
进队后再出队列的！

43. 判定一个顺序栈 st（最多元素为 MaxSize）为空的条件是\_\_B\_\_。

- A. st->top != -1    B. st->top == -1  
C. st->top != MaxSize    D. st->top == MaxSize

44. 判定一个顺序栈 st（最多元素为 MaxSize）为满的条件是\_\_D\_\_。

- A. st->top != -1    B. st->top == -1  
C. st->top != MaxSize    D. st->top == MaxSize

45. 一个队列的入队序列是 1, 2, 3, 4，则队列的输出序列是\_\_B\_\_。

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

FIFO

46. 判定一个循环队列 qu（最多元素为 MaxSize）为空的条件是\_\_C\_\_。

- A. qu->rear - qu->front == MaxSize    B. qu->rear - qu->front - 1 == MaxSize  
C. qu->rear == qu->front    D. qu->rear == qu->front - 1

47. 在循环队列中，若 front 与 rear 分别表示队头元素和队尾元素的位置，则判断循环队列空的条件是\_\_C\_\_。

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

48. 向一个栈顶指针为 h 的带头结点的链栈中插入指针 s 所指的结点时，应执行\_\_D\_\_操作。

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

49. 输入序列为 ABC，可以变为 CBA 时，经过的栈操作为\_\_B\_\_。

- A. push, pop, push, pop, push, pop    B. push, push, push, pop, pop, pop  
C. push, push, pop, pop, push, pop    D. push, pop, push, push, pop, pop

50. 若栈采用顺序存储方式存储，现两栈共享空间 V[1..m]，top[1]、top[2] 分别代表第 1 和第 2 个栈的栈顶，栈 1 的底在 V[1]，栈 2 的底在 V[m]，则栈满的条件是\_\_B\_\_。

- A. |top[2] - top[1]| = 0    B. top[1] + 1 = top[2]    C. top[1] + top[2] = m    D. top[1] = top[2]

top 指向栈顶 !!!

51. 设计一个判别表达式中左、右括号是否配对出现的算法，采用 D 数据结构最佳。  
A. 线性表的顺序存储结构      B. 队列      C. 线性表的链式存储结构      D. 栈
52. 允许对队列进行的操作有 D 。  
A. 对队列中的元素排序      B. 取出最近进队的元素  
C. 在队头元素之前插入元素      D. 删除队头元素
53. 对于循环队列 D 。  
A. 无法判断队列是否为空      B. 无法判断队列是否为满  
C. 队列不可能满      D. 以上说法都不对
54. 若用一个大小为 6 的数值来实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，当从队列中删除一个元素，再加入两个元素后，rear 和 front 的值分别为 B 。  
A. 1 和 5      B. 2 和 4      C. 4 和 2      D. 5 和 1
55. 队列的“先进先出”特性是指 D 。  
A. 最早插入队列中的元素总是最后被删除  
B. 当同时进行插入、删除操作时，总是插入操作优先  
C. 每当有删除操作时，总是要先做一次插入操作  
D. 每次从队列中删除的总是最早插入的元素
56. 和顺序栈相比，链栈有一个比较明显的优势是 A 。  
A. 通常不会出现栈满的情况      B. 通常不会出现栈空的情况  
C. 插入操作更容易实现      D. 删除操作更容易实现
57. 用不带头结点的单链表存储队列，其头指针指向队头结点，尾指针指向队尾结点，则在进行出队操作时 C 。  
A. 仅修改队头指针      B. 仅修改队尾指针  
C. 队头、队尾指针都可能要修改      D. 队头、队尾指针都要修改  
**一般只需修改队头指针，不过当队列里面只有一个元素时需要同时修改尾指针！！**
58. 若串 S= 'software'，其子串的数目是 B 。  
A. 8      B. 37      C. 36      D. 9  
 **$1+8+7+6+5+4+3+2+1=37$  !**
59. 串的长度是指 B 。  
A. 串中所含不同字母的个数      B. 串中所含字符的个数  
C. 串中所含不同字符的个数      D. 串中所含非空格字符的个数
60. 串是一种特殊的线性表，其特殊性体现在 B 。  
A. 可以顺序存储      B. 数据元素是一个字符  
C. 可以链式存储      D. 数据元素可以是多个字符
61. 设有两个串 p 和 q，求 q 在 p 中首次出现的位置的运算称为 B 。  
A. 连接      B. 模式匹配      C. 求子串      D. 求串长

**获得一个子串！**

62. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到 10, 从首地址 SA 开始连续存放的存储器内, 该数组按行存放, 元素 A[8][5]的起始地址为\_\_C\_\_。

A. SA+141 B. SA+144 C. SA+222 D. SA+225

$$A[i][j] \quad ((8-1)*10+(5-1))*3=222$$

63. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到 10, 从首地址 SA 开始连续存放的存储器内, 该数组按行存放, 元素 A[5][8]的起始地址为\_\_C\_\_。

A. SA+141 B. SA+180 C. SA+222 D. SA+225

个人和此答案不苟同! 此答案应为A!

64. 若声明一个浮点数数组如下: float average[]=new float[30];

假设该数组的内存起始位置为 200, average[15]的内存地址是\_\_C\_\_。

A. 214 B. 215 C. 260 D. 256

65. 设二维数组 A[1... m, 1... n]按行存储在数组 B 中, 则二维数组元素 A[i,j]在一维数组 B 中的下标为\_\_A\_\_。

A.  $n*(i-1)+j$  B.  $n*(i-1)+j-1$  C.  $i*(j-1)$  D.  $j*m+i-1$

66. 有一个  $100 \times 90$  的稀疏矩阵, 非 0 元素有 10, 设每个整型数占 2 个字节, 则用三元组表示该矩阵时, 所需的字节数是\_\_B\_\_。

A. 20 B. 66 C. 18 000 D. 33

$$m, n, nz, 10*3$$

67. 数组 A[0 ... 4, -1 ... -3, 5 ... 7]中含有的元素个数是\_\_A\_\_。

A. 55 B. 45 C. 36 D. 16

68. 对矩阵进行压缩存储是为了\_\_D\_\_。

A. 方便运算 B. 方便存储 C. 提高运算速度 D. 减少存储空间

69. 设有一个 10 阶的对称矩阵 A, 采用压缩存储方式, 以行序为主存储,  $a_{1,1}$  为第一个元素, 其存储地址为 1, 每个元素占 1 个地址空间, 则  $a_{8,5}$  的地址为\_\_B\_\_。

A. 13 B. 33 C. 18 D. 40

$$(1+2+3+4+5+6+7+5)$$

70. 稀疏矩阵一般的压缩存储方式有两种, 即\_\_C\_\_。

A. 二维数组和三维数组 B. 三元组和散列  
C. 三元组和十字链表 D. 散列和十字链表

71. 树最适合用来表示\_\_C\_\_。

A. 有序数据元素 B. 无序数据元素  
C. 元素之间具有分支层次关系的数据 D. 元素之间无联系的数据

72. 深度为 5 的二叉树至多有\_\_C\_\_个结点。

A. 16 B. 32 C. 31 D. 10

$$2^k - 1$$

73. 对一个满二叉树, m 个叶子, n 个结点, 深度为 h, 则\_\_D\_\_。

A.  $n = h+m$  B.  $h+m = 2n$  C.  $m = h-1$  D.  $n = 2^h - 1$



74. 任何一棵二叉树的叶子结点在前序、中序和后序遍历序列中的相对次序 A。

A. 不发生改变 B. 发生改变 C. 不能确定 D. 以上都不对

75. 在线索化树中，每个结点必须设置一个标志来说明它的左、右链指向的是树结构信息，还是线索化信息，若 0 标识树结构信息，1 标识线索，对应叶结点的左右链域，应标识为 D。

A. 00 B. 01 C. 10 D. 11

76. 在下述论述中，正确的是 D。

最大为2

有序!!!

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 的顺序二叉树的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

77. 设森林 F 对应的二叉树为 B，它有 m 个结点，B 的根为 p，p 的右子树的结点个数为 n，森林 F 中第一棵树的结点的个数是 A。

A. m-n B. m-n-1 C. n+1 D. 不能确定

78. 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 的结点的个数是 B。

A. 9 B. 11 C. 15 D. 不能确定

$n_0 = n_2 + 1$

79. 具有 10 个叶子结点的二叉树中有 B 个度为 2 的结点。

A. 8 B. 9 C. 10 D. 11

80. 在一个无向图中，所有顶点的度数之和等于所有边数的 C 倍。

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

81. 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 B 倍。

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

一个出度对应一个入度!

82. 某二叉树结点的中序序列为 ABCDEFG，后序序列为 BDCAFGE，则其左子树中结点数目为： C

A. 3 B. 2 C. 4 D. 5

83. 已知一算术表达式的中缀形式为  $A+B * C - D / E$ ，后缀形式为  $ABC * + DE / -$ ，其前缀形式为 D。

A.  $-A+B * C / DE$  B.  $-A+B * CD / E$  C.  $-+ * ABC / DE$  D.  $+A * BC / DE$

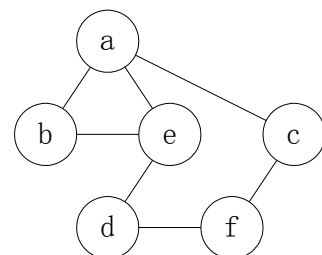
84. 已知一个图，如图所示，若从顶点 a 出发按深度搜索法进行遍历，则可能得到的一种顶点序列为 D；按广度搜索法进行遍历，则可能得到的一种顶点序列为 A；

①A. a, b, e, c, d, f B. a, c, f, e, b, d

C. a, e, b, c, f, d, D. a, e, d, f, c, b

②A. a, b, c, e, d, f B. a, b, c, e, f, d

C. a, e, b, c, f, d, D. a, c, f, d, e, b



此题应为B，每次访问都是依据上次访问的节点，从他们的邻接点开始访问下一层。这里c在e之前，所以f在d之前！



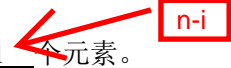
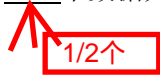

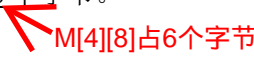
85. 采用邻接表存储的图的深度优先遍历算法类似于二叉树的 A。
- A. 先序遍历      B. 中序遍历      C. 后序遍历      D. 按层遍历
86. 采用邻接表存储的图的广度优先遍历算法类似于二叉树的 D。
- A. 先序遍历      B. 中序遍历      C. 后序遍历      D. 按层遍历
87. 具有  $n$  个结点的连通图至少有 A 条边。
- A.  $n-1$       B.  $n$       C.  $n(n-1)/2$       D.  $2n$
88. 广义表  $((a), a)$  的表头是 C，表尾是 C。
- A.  $a$       B.  $()$       C.  $(a)$       D.  $((a))$
- 表尾必须为列表！
89. 广义表  $((a))$  的表头是 C，表尾是 B。
- A.  $a$       B.  $()$       C.  $(a)$       D.  $((a))$
90. 顺序查找法适合于存储结构为 B 的线性表。
- A. 散列存储      B. 顺序存储或链式存储      C. 压缩存储      D. 索引存储
91. 对线性表进行折半查找时，要求线性表必须 B。
- A. 以顺序方式存储      B. 以顺序方式存储，且结点按关键字有序排列
- C. 以链式方式存储      D. 以链式方式存储，且结点按关键字有序排列
92. 采用折半查找法查找长度为  $n$  的线性表时，每个元素的平均查找长度为 D。
- A.  $O(n^2)$       B.  $O(n\log_2 n)$       C.  $O(n)$       D.  $O(\log_2 n)$
93. 有一个有序表为  $\{1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100\}$ ，当折半查找值为 82 的结点时，C 次比较后查找成功。
- A. 11      B. 5      C. 4      D. 8
- 7,10,12,11
94. 二叉树为二叉排序树的充分必要条件是任一结点的值均大于其左孩子的值、小于其右孩子的值。这种说法 B。
- A. 正确      B. 错误
95. 下面关于 B 树和 B+树的叙述中，不正确的结论是 A。
- A. B 树和 B+树都能有效的支持顺序查找      B. B 树和 B+树都能有效的支持随机查找
- C. B 树和 B+树都是平衡的多叉树      D. B 树和 B+树都可用于文件索引结构
96. 以下说法错误的是 B。
- A. 散列法存储的思想是由关键字值决定数据的存储地址
- B. 散列表的结点中只包含数据元素自身的信息，不包含指针。
- C. 负载因子是散列表的一个重要参数，它反映了散列表的饱满程度。
- D. 散列表的查找效率主要取决于散列表构造时选取的散列函数和处理冲突的方法。

97. 查找效率最高的二叉排序树是 C。
- A. 所有结点的左子树都为空的二叉排序树。  
 B. 所有结点的右子树都为空的二叉排序树。  
 C. 平衡二叉树。  
 D. 没有左子树的二叉排序树。
98. 排序方法中, 从未排序序列中依次取出元素与已排序序列中的元素进行比较, 将其放入已排序序列的正确位置上的方法, 称为 C。
- A. 希尔排序      B. 冒泡排序      C. 插入排序      D. 选择排序
99. 在所有的排序方法中, 关键字比较的次数与记录的初始排列次序无关的是 D。
- A. 希尔排序      B. 冒泡排序      C. 直接插入排序      D. 直接选择排序
100. 堆是一种有用的数据结构。下列关键码序列 D 是一个堆。
- A. 94,31,53,23,16,72      B. 94,53,31,72,16,23  
 C. 16,53,23,94,31,72      D. 16,31,23,94,53,72
101. 堆排序是一种 B 排序。
- A. 插入      B. 选择      C. 交换      D. 归并
102. D 在链表中进行操作比在顺序表中进行操作效率高。
- A. 顺序查找      B. 折半查找      C. 分块查找      D. 插入
103. 直接选择排序的时间复杂度为 D。(n 为元素个数)
- A.  $O(n)$       B.  $O(\log_2 n)$       C.  $O(n \log_2 n)$       D.  $O(n^2)$

$$\left\{ \begin{array}{l} k_i \leq k_{2i} \\ k_i \leq k_{2i+1} \end{array} \right. \quad \text{或者} \quad \left\{ \begin{array}{l} k_i \geq k_{2i} \\ k_i \geq k_{2i+1} \end{array} \right.$$

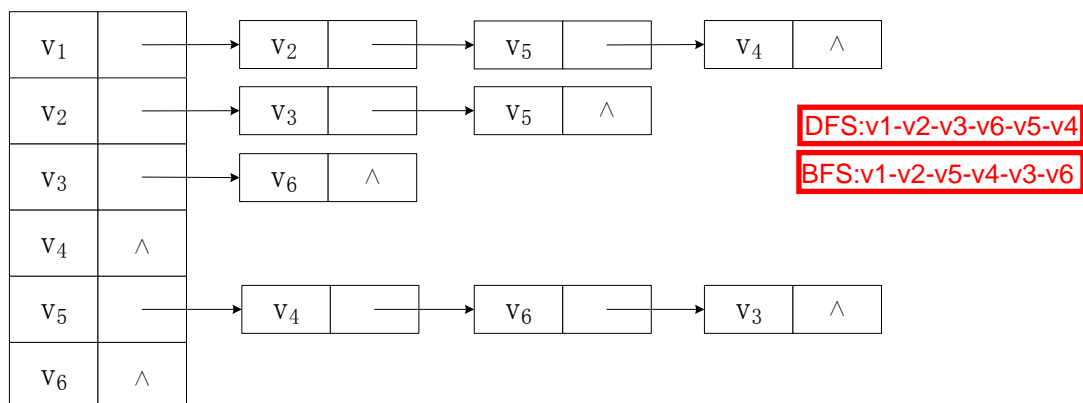
## 二、填空题。

- 数据逻辑结构包括 线性结构、树形结构 和 图状结构 三种类型, 树形结构和图状结构合称 非线性结构。
- 数据的逻辑结构分为 集合、线性结构、树形结构 和 图状结构 4 种。
- 在线性结构中, 第一个结点 没有 前驱结点, 其余每个结点有且只有 1 个前驱结点; 最后一个结点 没有 后续结点, 其余每个结点有且只有 1 个后续结点。
- 线性结构中元素之间存在 一对一 关系, 树形结构中元素之间存在 一对多 关系, 图形结构中元素之间存在 多对多 关系。
- 在树形结构中, 树根结点没有 前驱 结点, 其余每个结点有且只有 1 个前驱结点; 叶子结点没有 后续 结点, 其余每个结点的后续结点可以 任意多个。

6. 数据结构的基本存储方法是顺序、链式、索引和散列存储。
7. 衡量一个算法的优劣主要考虑正确性、可读性、健壮性和时间复杂度与空间复杂度。
8. 评估一个算法的优劣，通常从时间复杂度和空间复杂度两个方面考察。
9. 算法的5个重要特性是有穷性、确定性、可行性、输入和输出。
10. 在一个长度为  $n$  的顺序表中删除第  $i$  个元素时，需向前移动  $n-i-1$  个元素。
11. 在单链表中，要删除某一指定的结点，必须找到该结点的前驱结点。
12. 在双链表中，每个结点有两个指针域，一个指向前驱结点，另一个指向后继结点。
13. 在顺序表中插入或删除一个数据元素，需要平均移动  $n$  个数据元素，移动数据元素的个数与位置有关。
14. 当线性表的元素总数基本稳定，且很少进行插入和删除操作，但要求以最快的速度存取线性表的元素是，应采用顺序存储结构。
15. 根据线性表的链式存储结构中每一个结点包含的指针个数，将线性链表分成单链表和双链表。
16. 顺序存储结构是通过下标表示元素之间的关系；链式存储结构是通过指针表示元素之间的关系。
17. 带头结点的循环链表  $L$  中只有一个元素结点的条件是  $L \rightarrow next \rightarrow next = L$ 。
18. 栈是限定仅在表尾进行插入或删除操作的线性表，其运算遵循后进先出的原则。
19. 空串是零个字符的串，其长度等于零。空白串是由一个或多个空格字符组成的串，其长度等于其包含的空格个数。
20. 组成串的数据元素只能是单个字符。
21. 一个字符串中任意个连续字符构成的部分称为该串的子串。
22. 子串 "str" 在主串 "datastructure" 中的位置是5。
23. 二维数组  $M$  的每个元素是6个字符组成的串，行下标  $i$  的范围从0到8，列下标  $j$  的范围从1到10，则存放  $M$  至少需要 540 个字节； $M$  的第8列和第5行共占 108 个字节。
24. 稀疏矩阵一般的压缩存储方法有两种，即三元组表和十字链表。

25. 广义表  $((a), ((b), c), (((d))))$  的长度是 3，深度是 4。
26. 在一棵二叉树中，度为零的结点的个数为  $n_0$ ，度为 2 的结点的个数为  $n_2$ ，则有  $n_0 = \underline{n_2 + 1}$ 。
27. 在有  $n$  个结点的二叉链表中，空链域的个数为  $n+1$ 。
28. 一棵有  $n$  个叶子结点的哈夫曼树共有  $2n-1$  个结点。 2011年5月20日0:31
29. 深度为 5 的二叉树至多有 31 个结点。
30. 若某二叉树有 20 个叶子结点，有 30 个结点仅有一个孩子，则该二叉树的总结点个数为 69。  
 $20+30+19=69$
31. 某二叉树的前序遍历序列是  $abdgcefh$ ，中序序列是  $dgbacfh$ ，其后序序列为  $gdbefhca$ 。
32. 线索二叉树的左线索指向其 遍历序列中的前驱，右线索指向其 遍历序列中的后继。
33. 在各种查找方法中，平均查找长度与结点个数  $n$  无关的查找方法是 散列查找法。 散列查找法！！
34. 在分块索引查找方法中，首先查找 索引表，然后查找相应的 块表。
35. 一个无序序列可以通过构造一棵 二叉排序 树而变成一个有序序列，构造树的过程即为对无序序列进行排序的过程。
36. 具有 10 个顶点的无向图，边的总数最多为 45。  
 $(n-1)*n/2$  !

37. 已知图 G 的邻接表如图所示，其从顶点  $v_1$  出发的深度优先搜索序列为  $v_1v_2v_3v_6v_5v_4$ ，其从顶点  $v_1$  出发的广度优先搜索序列为  $v_1v_2v_5v_4v_3v_6$ 。



38. 索引是为了加快检索速度而引进的一种数据结构。一个索引隶属于某个数据记录集，它由若干索引项组成，索引项的结构为 关键字 和 关键字对应记录的地址。

$e \leq n-1$   
无回路产生  
当前所选择的边的权值最小

39. Prim 算法生成一个最小生成树每一步选择都要满足 边的总数不超过  $n-1$  ,  
当前选择的边的权值是候选边中最小的 , 选中的边加入树中不产生回路 三项原则。

40. 在一棵  $m$  阶 B 树中, 除根结点外, 每个结点最多有  $m$  棵子树, 最少有  $m/2$  棵子树。

### 三、判断题。

数据单一的话可以选用顺序存储结构！

1. 在决定选取何种存储结构时, 一般不考虑各结点的值如何。(✓)
2. 抽象数据类型 (ADT) 包括定义和实现两方面, 其中定义是独立于实现的, 定义仅给出一个 ADT 的逻辑特性, 不必考虑如何在计算机中实现。(✓)
3. 抽象数据类型与计算机内部表示和实现无关。(✓)
4. 顺序存储方式插入和删除时效率太低, 因此它不如链式存储方式好。(×) 各有所好！！
5. 线性表采用链式存储结构时, 结点和结点内部的存储空间可以是不连续的。(×) 结构体, 连续！
6. 对任何数据结构链式存储结构一定优于顺序存储结构。(×)
7. 顺序存储方式只能用于存储线性结构。(×)
8. 集合与线性表的区别在于是否按关键字排序。(×) 集合无序, 线性表有序！
9. 线性表中每个元素都有一个直接前驱和一个直接后继。(×)
10. 线性表就是顺序存储的表。(×)
11. 取线性表的第  $i$  个元素的时间同  $i$  的大小有关。(×)
12. 循环链表不是线性表。(×)
13. 链表是采用链式存储结构的线性表, 进行插入、删除操作时, 在链表中比在顺序表中效率高。(✓)
14. 双向链表可随机访问任一结点。(×)
15. 在单链表中, 给定任一结点的地址  $p$ , 则可用下述语句将新结点  $s$  插入结点  $p$  的后面 :  
 $p \rightarrow next = s; s \rightarrow next = p \rightarrow next;$  (×)  $s \rightarrow next = p \rightarrow next; p \rightarrow next = s;$
16. 队列是一种插入和删除操作分别在表的两端进行的线性表, 是一种先进后出的结构。(×) 先进先出！  
数据元素是单个字符！
17. 串是一种特殊的线性表, 其特殊性体现在可以顺序存储。(×)
18. 长度为 1 的串等价于一个字符型常量。(×)
19. 空串和空白串是相同的。(×) 空白串存储的为空白字符, 空串为空, 不存储任何东西！
20. 数组元素的下标值越大, 存取时间越长。(×) 随机存储！
21. 用邻接矩阵法存储一个图时, 在不考虑压缩存储的情况下, 所占用的存储空间大小只与图中结点个数有关, 而与图的边数无关。(✓)
22. 一个广义表的表头总是一个广义表。(×)
23. 一个广义表的表尾总是一个广义表。(✓)
24. 广义表  $((a), b, c)$  的表头是  $((a), b)$ , 表尾是  $(c)$ 。(✓)
25. 二叉树的后序遍历序列中, 任意一个结点均处在其孩子结点的后面。(✓)
26. 度为 2 的有序树是二叉树。(×)
27. 二叉树的前序遍历序列中, 任意一个结点均处在其孩子结点的前面。(✓)
28. 用一维数组存储二叉树时, 总是以前序遍历顺序存储结点。(×) 层次遍历！
29. 若已知一棵二叉树的前序遍历序列和后序遍历序列, 则可以恢复该二叉树。(×) 必须要有中序！
30. 在哈夫曼树中, 权值最小的结点离根结点最近。(×) 远！
31. 强连通图的各项点间均可达。(✓)
32. 对于任意一个图, 从它的某个结点进行一次深度或广度优先遍历可以访问到该图的每个顶点。(×) XXXXX连通? 有向, 强连通?

33. 在待排序的记录集中, 存在多个具有相同键值的记录, 若经过排序, 这些记录的相对次序仍然保持不变, 称这种排序为稳定排序。(√)
34. 在平衡二叉树中, 任意结点左右子树的高度差(绝对值)不超过 1。(√)
35. 拓扑排序是按 AOE 网中每个结点事件的最早发生时间对结点进行排序。(×)
36. 冒泡排序算法关键字比较的次数与记录的初始排列次序无关。(×) **有关, 因为如果某此遍历无交换数据则表明排序完成!**
37. 对线性表进行折半查找时, 要求线性表必须以链式方式存储, 且结点按关键字有序排列。(×) **顺序**
38. 散列法存储的思想是由关键字值决定数据的存储地址。(√) **左孩子子树的所有值, 小于右孩子子树ALL!**
39. 二叉树为二叉排序树的充分必要条件是任一结点的值均大于其左孩子的值、小于其右孩子的值。(×)
40. 具有  $n$  个结点的二叉排序树有多种, 其中树高最小的二叉排序树是最佳的。(√) **深度最小!**
41. 直接选择排序算法在最好情况下的时间复杂度为  $O(n)$ 。(×)

#### 四、应用简答题。

**a-->b->c-->d-->e-->f-->g-->h**

1. 有下列几种用二元组表示的数据结构, 画出它们分别对应的逻辑图形表示, 并指出它们分别属于何种结构。

(1)  $A = \{D, R\}$ , 其中:  $D = \{a, b, c, d, e, f, g, h\}$ ,  $R = \{r\}$  **线性存储结构!**

$r = \{<a, b>, <b, c>, <c, d>, <d, e>, <e, f>, <f, g>, <g, h>\}$

(2)  $B = \{D, R\}$ , 其中:  $D = \{a, b, c, d, e, f, g, h\}$ ,  $R = \{r\}$ ,

$r = \{<d, b>, <d, g>, <d, a>, <b, c>, <g, e>, <g, h>, <e, f>\}$  **树形结构**

(3)  $C = \{D, R\}$ , 其中:  $D = \{1, 2, 3, 4, 5, 6\}$ ,  $R = \{r\}$ ,

$r = \{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$  **图形结构**

**三种存储结构:**  
1. 线性结构  
2. 树形结构  
3. 图形结构

2. 简述顺序表和链表存储方式的特点。

答: 顺序表的优点是可以随机访问数据元素, 缺点是大小固定, 不利于增减结点(增减结点操作需要移动元素)。链表的优点是采用指针方式增减结点, 非常方便(只需改变指针指向, 不移动结点)。其缺点是不能进行随机访问, 只能顺序访问。另外, 每个结点上增加指针域, 造出额外存储空间增大。

3. 对链表设置头结点的作用是什么?(至少说出两条好处)

答: 其好处有:

(1) 对带头结点的链表, 在表的任何结点之前插入结点或删除表中任何结点, 所要做的都是修改前一个结点的指针域, 因为任何元素结点都有前驱结点(若链表没有头结点, 则首元素结点没有前驱结点, 在其前插入结点和删除该结点时操作复杂些)。

(2) 对带头结点的链表, 表头指针是指向头结点的非空指针, 因此空表与非空表的处理是一样的。

**头结点的作用:**  
1. 空表与非空表处理一样  
2. 结点之前操作更方便。

4. 对于一个栈, 给出输入项 A, B, C。如果输入项序列由 A, B, C 组成, 试给出全部可能的输出序列。

**ABC ACB BAC BCA ~~CAB~~ CBA 共5种!**

5. 设有 4 个元素 1、2、3、4 依次进栈, 而栈的操作可随时进行(进出栈可任意交错进行, 但要保证进栈次序不破坏 1、2、3、4 的相对次序), 请写出所有不可能的出栈次序和所有可能的出栈次序。

6. 现有稀疏矩阵 A 如图所示，要求画出三元组表示法和十字链表表示法：

|    |    |    |    |   |     |                                                                                 |
|----|----|----|----|---|-----|---------------------------------------------------------------------------------|
| 15 | 0  | 0  | 22 | 0 | -15 | 三元组：{ (1,1,15),(1,4,22),(1,6,-15),(2,2,13),(2,3,3),(3,4,-6),(5,1,91),(6,3,28) } |
| 0  | 13 | 3  | 0  | 0 | 0   | 15--->22-->15                                                                   |
| 0  | 0  | 0  | -6 | 0 | 0   | 13--->3                                                                         |
| 0  | 0  | 0  | 0  | 0 | 0   | -6--->0                                                                         |
| 0  | 0  | 0  | 0  | 0 | 0   | 00--->0                                                                         |
| 91 | 0  | 0  | 0  | 0 | 0   | 91--->0                                                                         |
| 0  | 0  | 28 | 0  | 0 | 0   | 28--->0                                                                         |

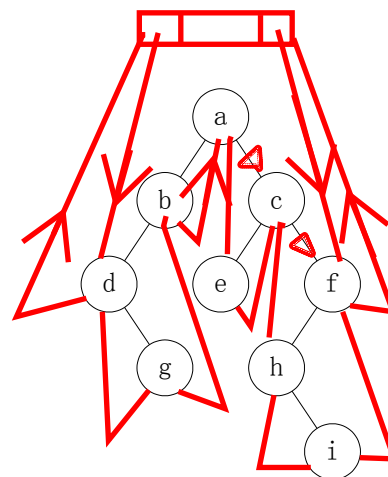
7. 设 4 维数组的 4 个下标的范围分别为

[-1, 0], [1, 2], [1, 3], [-2, -1]，请分别按行序和列序列出各元素。

8. 有一份电文中共使用 5 个字符：a, b, c, d, e，它们出现的频率依次为 4, 7, 5, 2, 9，试画出对应的哈夫曼树（请按左子树根结点的权小于等于右子树根结点的权的次序构造），并求出每个字符的哈夫曼编码。

9. 有如图所示的二叉树，回答如下问题。

- (1) 写出该树的中序遍历序列：d,g,b,a,e,c,h,i,f
- (2) 写出该树的先序遍历序列：a,b,d,g,c,e,f,h,i
- (3) 写出该树的后序遍历序列：g,d,b,e,i,h,f,c,a
- (4) 画出该二叉树的中序线索二叉树；
- (5) 画出该二叉树的后序线索二叉树；
- (6) 画出该二叉树对应的森林；



10. 已知一棵树边 的 集 合 为 {<i,m>,<i,n>,<e,i>,<b,e>,<b,d>,<a,b>,<g,j>,<g,k>,<c,g>,<c,f>,<h,l>,<c,h> ,<a,c>}，画出这棵树。

11. 假设二叉树采用顺序存储结构，如图所示。

- (1) 画出二叉树表示；
- (2) 写出先序遍历、中序遍历和后序遍历的结果；
- (3) 写出结点值 c 的双亲结点，其左、右孩子；
- (4) 画出把此二叉树还原成森林的图。

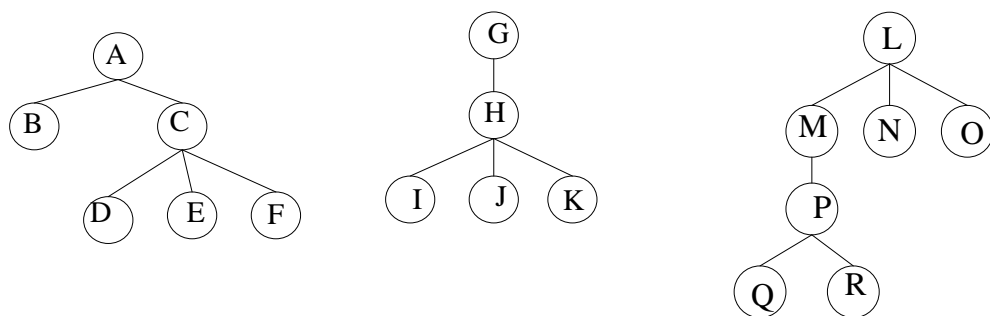
|   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| e | a | f |   | d |   | g |   |   | c  | j  |    |    | h  | i  |    |    |    |    | b  |

12. 已知一棵二叉树的中序序列为 cbedahgijf，后序序列为 cedbhjigfa，画出该二叉树的先序线索二叉树。



13. 某二叉树的先序遍历序列是  $abdgcefh$ ，中序遍历序列是  $dgbaechf$ ，给出其后序遍历序列。

14. 将下图所示森林转换成为二叉树，并写出转化后二叉树中序遍历结果。



15. 有一份电文中共使用 8 个字符：a、b、c、d、e、f、o、i，它们的出现频率依次为 10，20，15，32，40，60，26，18。试画出对应的哈夫曼树（请按左子树根结点的权小于等于右子树根结点的权的次序构造），并求出每个字符的哈夫曼编码。

16. 已知某系统在通信联络中只可能出现 A,B,C,D,E,F,G,H 八种字符，其频率为 0.05，0.29，0.07，0.08，0.14，0.23，0.03，0.11 试设计哈夫曼编码。

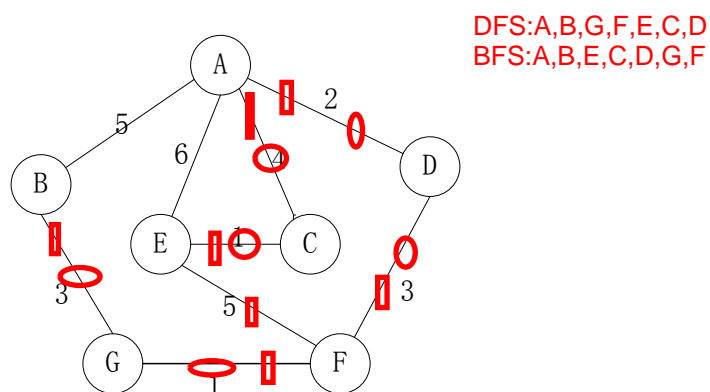
17. 对有五个顶点  $\{v_1, v_2, v_3, v_4, v_5\}$  的图的邻接矩阵如图所示，解答下列问题：

- (1) 画出逻辑图。
- (2) 画出该逻辑结构的邻接表。
- (3) 基于邻接矩阵写出图的深度、广度优先遍历序列。

$$\begin{bmatrix} 0 & 100 & 30 & \infty & 10 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 60 & 0 & 20 & \infty \\ \infty & 10 & \infty & 0 & \infty \\ \infty & \infty & \infty & 50 & 0 \end{bmatrix}$$

18. 如图所示，解答如下问题：

- (1) 写出从定点 A 出发，深度和广度优先遍历方法遍历该图的顶点序列。
- (2) 根据普里姆算法和克鲁斯卡尔算法，分别求它的最小生成树，要求给出构造过程。



19. 给出如图所示的无向图 G 的邻接矩阵和邻接表两种存储结构。并在给定的邻接表的基础上，指出从顶点 1 出发的深度优先遍历和广度优先遍历序列。

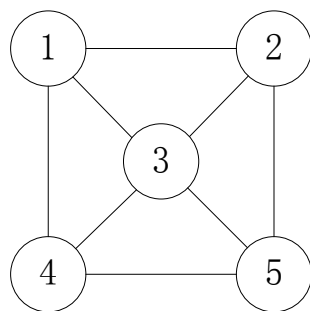


图 一个无向图G

20. 使用普里姆算法构造出如图所示的图 G 的一棵最小生成树。

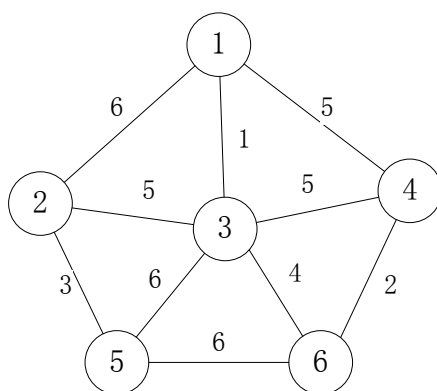


图 一个无向图G

21. 使用克鲁斯卡尔算法构造出如图所示的图 G 的一棵最小生成树。

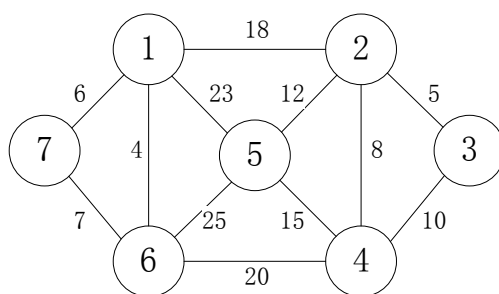


图 一个无向图G

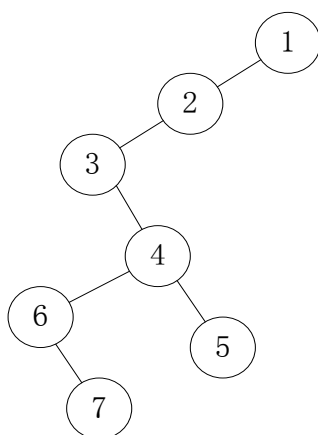
22. 设有一棵二叉树，它的中序和后序遍历结果如下，请画出该二叉树。

中序：1 4 3 5 6 2

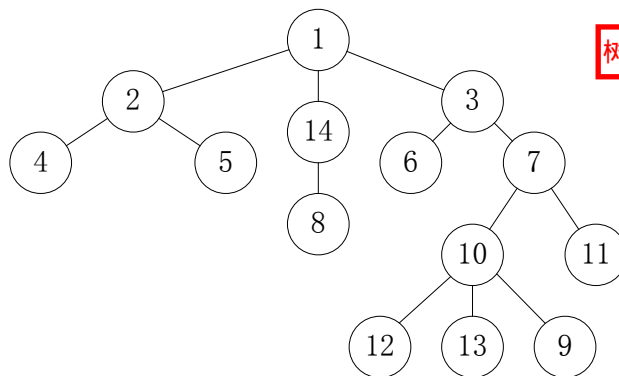
后序：4 6 5 3 2 1

23. 设一棵顺序二叉树具有 10 个结点，请计算其中叶子结点的数目。

24. 设如图所示二叉树是由某棵树转化而来，请画出其对应的原树。



25. 设有如图所示的一棵树，请将其转化为二叉树。



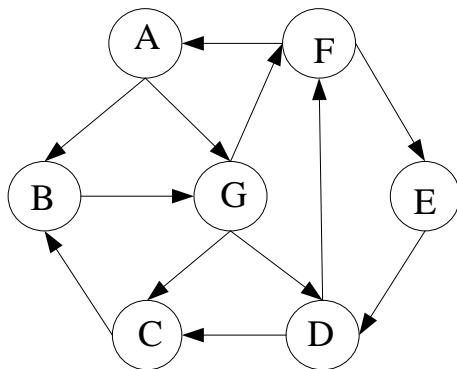
树的孩子兄弟表示法！

26. 下表给出了某工程各工序之间的优先关系和各工序所需时间。解答下列问题：

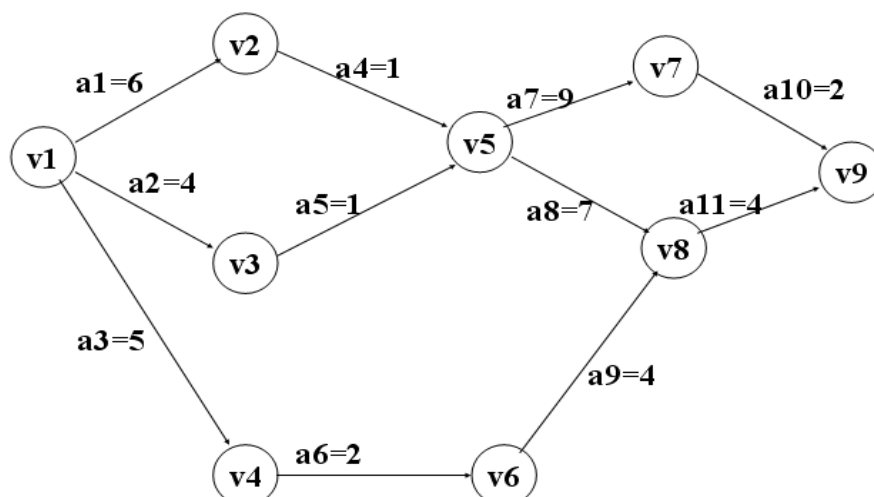
- (1) 画出相应的 AOE 图；
- (2) 给出各事件的最早发生时间和最晚发生时间；
- (3) 找出关键路径，并指明完成该工程所需最短时间；
- (4) 若把 AOE 网视为 AOV 网，给出其一个拓扑序列的例子。

| 工 序<br>代号 | A  | B  | C   | D | E   | F  | G  | H   | I  | J  | K   | L       | M  | M  |
|-----------|----|----|-----|---|-----|----|----|-----|----|----|-----|---------|----|----|
| 时间        | 15 | 10 | 50  | 8 | 15  | 40 | 90 | 15  | 80 | 60 | 15  | 30      | 20 | 40 |
| 先 驱<br>工作 | —  | —  | A,B | B | C,D | B  | E  | G,I | E  | I  | F,I | H,J,K,L | M  | G  |

27. 某不带权有向图如下所示。给出其邻接矩阵和邻接表表示。



28. 求如下 AOE 图的关键路径，要求给出求解过程。



29. 有一组数据，内容如下：8, 15, 38, 57, 68, 88, 98, 108, 129, 234, 256  
试用二分查找法查找 68 和 222，要求先画出二叉折半检索树，然后写出查找过程。
30. 已知有序表为 {12, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134}，请画出采用折半查找法对应的判断树。
31. 设数据集  $d = \{1, 12, 5, 8, 3, 10, 7, 13, 9\}$ ，试完成下列各题：  
(1) 依次取  $d$  中各数据，构造一棵二叉排序树  $bt$ 。  
(2) 如何依据此二叉树  $bt$  得到  $d$  的一个有序序列。  
(3) 画出在二叉树  $bt$  中删除“12”后的树结构。
32. 对给定的数列  $R = \{7, 16, 4, 8, 20, 9, 6, 18, 5\}$ ，构造一棵二叉排序树，并且  
(1) 给出按中序遍历得到的数列  $R_1$ 。  
(1) 给出按后序遍历得到的数列  $R_2$ 。
33. 已知序列 {17, 18, 60, 40, 7, 32, 73, 65, 85}，请给出采用冒泡排序法对该序列作升序排序时每一趟的结果。
34. 已知序列 {503, 87, 512, 61, 908, 170, 897, 275, 653, 462}，请给出采用快速排序法对该序列作升序排序时每一趟的结果。
35. 已知序列 {503, 87, 512, 61, 908, 170, 897, 275, 653, 462}，请给出采用堆排序法对该序列作升序排序时每一趟的结果。
36. 已知序列 {503, 87, 512, 61, 908, 170, 897, 275, 653, 462}，请给出采用希尔排序法对该序列作升序排序时每一趟的结果。
37. 已知序列 {17, 18, 60, 40, 7, 32, 73, 65, 85}，请给出采用直接插入排序法对该序列作升序排序时每一趟的结果。
38. 设散列表的长度  $m = 13$  (0, 1, 2, ..., 12)，散列函数为  $H(k) = k \bmod m$ ，给定的关键字序列为 {19, 14, 23, 10, 68, 20, 84, 27, 55, 11}。试画出用线性探测法解决冲突时所构造的散列表。

## 五、算法设计题。

1. 已知一个顺序表 L，其中的元素按值非递减有序排列，设计一个算法插入一个元素 x 后保持该顺序表仍按非递减有序排列。  
**先扫描一遍，把L中x的下标记录在数组A[j]中！接着用循环来移动元素即可。**  
`for( i=1,j=A[0];j<N;j++) { if( A[j]<A[i] ) A[j]=A[j+i]; else i++; }`
2. 设计一个算法从顺序表 L 中删除所有值为 x 的元素。  
`else i++;`
3. 已知线性表元素递增有序，并以带头结点的单链表作存储结构，设计一个高效算法，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）。并分析所写算法的时间复杂度。  
`for( p=L;p->next->data<=mink;p=p->next ) ;//使得p->data<=minlink  
for( q=p;q->next->data<maxk;q = q->next ); p->next = q;`
4. 设计一个在带头结点的单链表中删除一个最小值结点的高效算法。
5. 有一个不带头结点的单链表 L（至少有一个结点），其头指针为 head。设计一个算法将 L 逆置，即最后一个结点变成第一个结点，原来倒数第二个结点变成第二个结点，如此等等。  
`for( p=L;p;p=p->next )  
if( p->next->data<min ) q=p;  
q->next = q->next->next;`
6. 假设二叉树采用链式存储方式存储，编写一个二叉树前序遍历的非递归算法。  
`Push(&s,p);while( !EmptyStack(s))  
{ while( GetTop(s,&p)&p ){ visit(p);  
p=p->lchild;Push(&s,p);}Pop(&s,&p);  
if( !EmptyStack(s) ){Pop(&s,&p);  
Push(&s,p->rchild);}}`
7. 假设二叉树采用链式存储方式存储，编写一个二叉树后序遍历的非递归算法。
8. 假设二叉树采用链式存储方式存储，编写一个二叉树中序遍历的非递归算法。
9. 编写一个 c++ 函数。实现线性表就地逆置。即在原表的存储空间内将线性表 (a1,a2,...,an) 逆置为 (an,...,a2,a1)。  
`mid=(1+n)/2;for( int i=1;i<=mid;i++ ) exchange(a[i],a[n-i+1] );`
10. 编写一个单链表倒链程序，即将单链表中每个结点的前驱与后继关系颠倒。
11. 在数组 a[0...n-1] 中存放有 n 个不同的整数，请编写一个函数，将 a 中的 n 个数按从小到大的顺序排列。
12. 有一个不带头结点的单链表 L（至少有一个结点），其头指针为 head。设计一个算法将 L 逆置，即最后一个结点变成第一个结点，原来倒数第二个结点变成第二个结点，如此等等。

13. 已知非空线性链表的第一个结点的指针为 head，请写一个算法，将该链表中数据域值最小的结点移动到链表的最前端。编写的函数具有如下原型：**void func(TLinkNode \*head)**，其中链结点的结构如下：

```
struct TLinkNode
{
 int data;
 TLinkNode *next;
}
```

请完成该算法。

14. 在数组 a[0...n-1] 中存放有 n 个不同的整数，请编写一个函数，将 a 中的 n 个数按从小到大的顺序排列，要求使用改进的插入排序算法，元素 a[i] 要插入的位置由折半（二分）查找算法找到。