数据结构复习题

一、单项选择题

1. 算法分析的目的是（ ）。

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 D.分析算法的易懂性和文档性

2. 计算机算法指的是（ ）。

A.计算方法 B. 排序方法 C.求解问题的有限运算序列 D.调度方法

3. 数据结构中，与所使用的计算机无关的是数据的什么结构（ ）。

A.存储 B. 物理 C. 逻辑 D. 物理和存储

4. 算法分析的两个主要方面是( )。

A. 空间复杂性和时间复杂性 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

5. 一个向量第一个元素的存储地址是100，每个元素的长度为2，则第5个元素的地址是( )。

A.110 B.108 C.100 D.120

6.数据的逻辑结构可以分为（ ）两类。

A.紧凑结构和非紧凑结构 B.内部结构和外部结构

C.线性结构和非线性结构 D.动态结构和静态结构

7.计算机中算法指的是解决某一问题的有限运算序列，它必须具备输入、输出、（ ）。

A.易读性、稳定性和确定性 B.可行性、可移植性和可扩充性

C.可行性、有穷性和确定性 D.确定性、有穷性和稳定性

8．顺序表具有的特点是（ ）。

A. 可随机访问任一元素　 B. 删除不需要移动元素

C. 不必事先预估存储空间 D. 插入不需要移动元素

9.下面程序的时间复杂为 （ ）。

for（i=1，s=0； i<=n； i++）{

t=1；

for(j=1；j<=i；j++) {

t=t\*j；

s=s+t；

}

}

A.O() B.O()

C.O(n) D.O()

10.关于线性表的正确说法是 （ ）。

A.线性表中至少有一个元素

B.每个元素都有一个前驱和一个后继元素

C.表中元素的排序顺序必须是由小到大或由大到小

D.除第一个元素和最后一个元素外，其余元素有且仅有一个前驱和一个后继元素

11.设顺序线性表中有 n 个数据元素，则删除表中第 i ()个元素需要移动（ ）个元素。

A.n-i B.n+l-i

C.n-1-i D.i

12．不带头结点的单链表head为空的判定条件是（ ）。

A. head==NULL B. head.next==NULL

C. head.next==head D. head!=NULL

13. 在单链表中查找指定值的节点的时间复杂度是（ ）。

A.O(log2n) B.O(1)

C.O() D.O(n)

14.在一个双链表中，在p节点之后插入节点q的操作是（ ）。

A.q.next = p.next; p.next.pre = q; p.next = q; q.pre = p;

B.p.next.pre = q; q.pre = p; p.next=q; q.next = p.next;

C.q.pre = p; p.next=q; p.next.pre = q; q.next = p.next;

D.p.next=q; q.pre = p; q.next = p.next; p.next.pre = q;

15．若一个线性表中最常用的操作是取第i个元素和找第i个元素的前驱元素，则采用（ ）存储方式最节省时间。

A.顺序表 B.单链表

C.双链表 D.单循环链表

16.串的长度是指（ ）

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

17.已知一个栈的进栈序列是ABC，出栈序列为CBA，经过栈的操作是（ ）。

A.push, pop, push, push, pop, pop

B.push, push, push, pop, pop, pop

C.push, pop, push, pop, push, pop

D.push, push, pop, pop, push, pop

18.设一个栈的输入序列为A、B、C、D，则借助一个栈所得的输出序列不可能是（ ）。

A.ABCD B.DCBA

C.DABC D.ACDB

19.表达式a\*(b+c)-d的后缀表达式是（ ）。

A.a b c + \* d - B.a b c d \* + -

C.a b c \* + d - D.- + \* a b c d

20.队列是一种（ ）的线性表。

A.先进先出 B.先进后出

C.只能插入 D.只能删除

21.对特殊矩阵采用压缩存储的目的主要是（ ）。

A.减少不必要的存储空间 B.表达变得简单

C.去掉矩阵中的多余元素 D.对矩阵元素的存储变得简单

22．设有一个包含n个元素的有序线性表。在等概率情况下删除其中的一个元素，若采用顺序存储结构，则平均需要移动（ ）个元素。

A.1 B.(n-1)/2

C.log2n D. n

23．算法分析的两个主要方面是（ ）。

A.空间复杂度和时间复杂度 B.正确性和简单性

C.可读性 D.数据复杂性和程序复杂性

24．下面的数据结构中，（ ）的元素之间存在多对多的关系。

A． 线性结构 B. 树结构

C. 图结构 D.集合结构

25. 在*n*个结点的顺序表中，算法的时间复杂度是O（1）的操作是（ ）。

A.访问第*i*个结点（1≤*i*≤*n*）和求第*i*（2≤*i*≤*n*）个结点的前驱结点

B.在第*i*（1≤*i*≤*n*）个结点后插入一个新结点

C.删除第*i*个结点（1≤*i*≤*n*）

D.将*n*个结点从小到大排序

26. 数据结构是为了研究（ ）。

A.数据的逻辑结构 B.数据的存储结构

C.数据的逻辑结构和存储结构 D.数据的逻辑结构、存储结构及其基本操作

27. 线性表若采用链式存储结构时，要求内存中可用存储单元的地址( )。

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.连续或不连续都可以

28. 线性表Ｌ在哪种情况下适用于使用链式结构实现。

A.需经常修改Ｌ中的结点值 B.需不断对Ｌ进行删除插入

C.Ｌ中含有大量的结点 D.Ｌ中结点结构复杂

29. 设一个栈的进栈序列是1、2、3、4（即元素1～4依次通过该栈），则借助该栈所得到的输出序列不可能是（ ）。

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

30.下面程序段的时间复杂度为（ ）。

for(i=0;i<m;i++)

for(j=0;j<n;j++)

a[i][j]=i\*j;

A. O(m2) B. O(n2)

C. O(m\*n) D. O(m+n)

31.下面的程序段不符合算法的（ ）原则。

void sam()

{

int n=8;

while(n>6)

n+=2;

printf(n);

}

A.健壮性 B.确定性

C.可行性 D.有穷性

32. 若已知一个栈的入栈序列是1，2，3，…，n，其输出序列为p1，p2，p3，…，pn，若p1=n，则pi为（ ）。

A.i B.n=i C.n-i+1 D.不确定

33. 若用一个大小为6的数组来实现循环队列，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的值分别为多少（）。

A.1和 5 B.2和4 C.4和2 D.5和1

34. 循环队列qu（队头指针front指向队首元素的前一位置，队尾指针rear指向队尾元素的位置）的队满条件是（ ）。

A. (qu.rear+1)%MaxSize==(qu.front+1)%MaxSize

B. (qu.rear+1)%MaxSize==qu.front+1

C.(qu.rear+1)%MaxSize==qu.front

D.qu.rear==qu.front

35. 以下（ ）是"*abcd321ABCD*"串的子串。

A. abcd B.321AB C."abcABC" D."21AB"

36. 假设有60行70列的二维数组*a*[1..60,1..70]（数组下标从1开始）以列序为主序顺序存储，其基地址为10000，每个元素占2个存储单元，那么第32行第58列的元素*a*[32,58]的存储地址为（ ）。（（58-1）\* 60 + 32-1）\* 2 + 10000

A.16902 B.16904 C.14454 D.以上都不对

37. 稀疏矩阵一般的压缩存储方法有两种，即（ ）。

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

38. 一个*n*×*n*的对称矩阵，如果采用压缩存储以行或列为主序放入内存，则压缩存储的容量是（ ）。

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

39. 若一棵度为7的树有7个度为2的结点，有6个度为3的结点，有5个度为4的结点，有4个度为5的结点，有3个度为6的结点，有2个度为7的结点，该树一共有（ ）个叶子结点。

2\*7+6\*3+5\*4+4\*5+3\*6+2\*7+1=105

105-7-6-5-4-3-2=78

A. 35 B. 28 C. 77 D. 78

40.下列陈述中正确的是( )。

A.二叉树中每个结点最多只有两棵子树，并且有左右之分

B.二叉树中必有度为2的结点

C.二叉树中结点只有一个孩子时无左右之分

D.二叉树是度为2的有序树

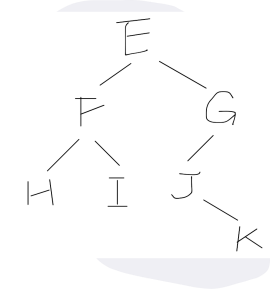
41.具有3个结点的二叉树有（ ）种形态。

A.3 B.5

C.4 D.6

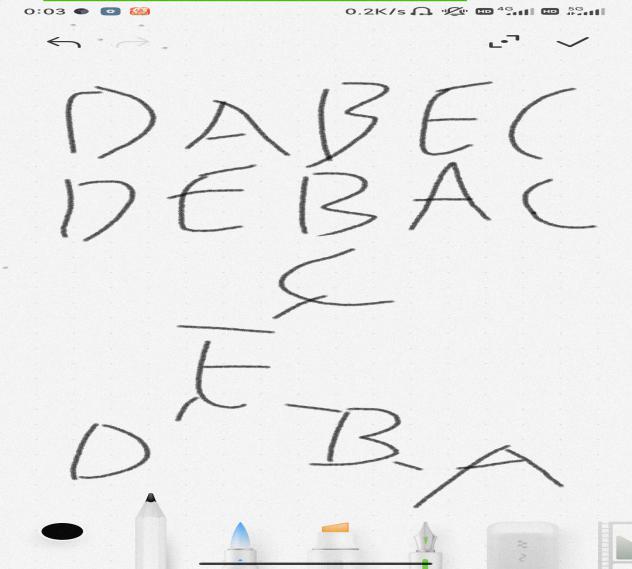
42.一棵二叉树的先序遍历序列为EFHIGJK,中序遍历序列为HFIEJKG,则该二叉树根结点的右孩子为（ ）。

A.H B.E

C.F D.G

43. 一棵二叉树的后序遍历序列为D、A、B、E、C，中序遍历序列为D、E、B、A、C，则先序遍历序列为（ ）。

A. A、C、B、E、D B. D、E、C、B、A

C. D、E、A、B、C D. C、E、D、B、A

44. 二叉树是非线性数据结构，所以( )。

A.它不能用顺序存储结构存储;

B.它不能用链式存储结构存储;

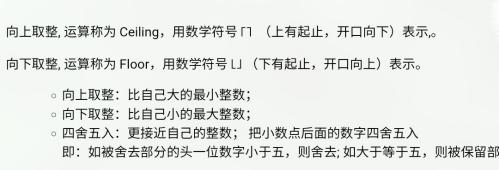
C.顺序存储结构和链式存储结构都能存储;

D.顺序存储结构和链式存储结构都不能使用

45. 具有n(n>0)个结点的完全二叉树的深度为（ ）。

A. ⎡ log2(n) ⎤ B. ⎣ log2(n) ⎦

C. ⎣ log2(n) ⎦ +1 D. ⎡ log2(n)+1 ⎤



46. 把一棵树转换为二叉树后，这棵二叉树的形态是（ ）。

A.唯一的 B.有多种

C.有多种，但根结点都没有左孩子 D.有多种，但根结点都没有右孩子

47. 设树T的度为4，其中度为1，2，3和4的结点个数分别为4，2，1，1 则T中的叶子数为（ ）。

1\*4+2\*2+3+4+1=16

16-4-2-1-1=8

A.5 B.6 C.7 D.8

48. 一棵树高为K的完全二叉树至少有多少个结点（ ）。 根据二叉树的性质，第1层到第k-1层共有结点2k-1-1个，因此它至少有2k-1-1+1=2k-1个结点。

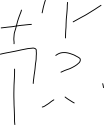
A.2（k –1） B. 2（k - 1）–1 C.2k -1 D.2k

49. 根据使用频率为5个字符设计的哈夫曼编码不可能是（ ）。度为0或2

A. 111,110,10,01,00 B. 000,001,010,011,1

C. 100,11,10,1,0 D. 001,000,01,11,10





50. 有8个顶点的无向图最多有（ ）条边。8\*7/2=28

A.14 B.28 C.56 D.112

51. 若一个图的邻接矩阵是对称矩阵，则该图一定是（ ）。

A.有向图 B.无向图 C.连通图 D.无向图或有向图

52. 在一个无向图中，所有顶点的度数之和等于图的边数的（ ）倍。

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

53. 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

54. 有8个结点的无向图最多有（ ）条边。n\*(n-1)/2 P195

A.14 B.28 C.56 D.112

55. 有8个结点的无向连通图最少有（ ）条边。n-1

A.5 B.6 C.7 D.8

56. 有8个结点的有向完全图有（ ）条边。n\*(n-1)

A.14 B.28 C.56 D.112

57. 用邻接表表示图进行广度优先遍历时，通常是采用（ ）来实现算法的。

A.栈 B.队列 C.树 D.图

58. 用邻接表表示图进行深度优先遍历时，通常是采用（ ）来实现算法的。

A.栈 B. 队列 C.树 D.图

59.设完全无向图中有 n 个顶点，则该完全无向图中有（ ）条边。

A.n(n-1)/2 B.n(n-1)

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

60.设有 6 个结点的无向图，该图至少应有（ ）条边才能是一个连通图。

A.5 B.6

C.7 D.8

61.对于线性表（7，34，55，25，64，46，20，10）进行散列存储时，若选用 H（K）=K%9 作为散列函数，则散列地址为 1 的元素有（ ）个。

A.1 B.2

C.3 D.4

62.对关键字序列（21,19,37,5,2），经直接插入排序法由小到大排序，第一趟后所得结果为（ ）。

A.(19，21，37，5，2) B.(19，21，5，2，37)

C.(19，21，5，37，2) D.(19，21，2，5，37)

63. 最小生成树指的是（ ）。

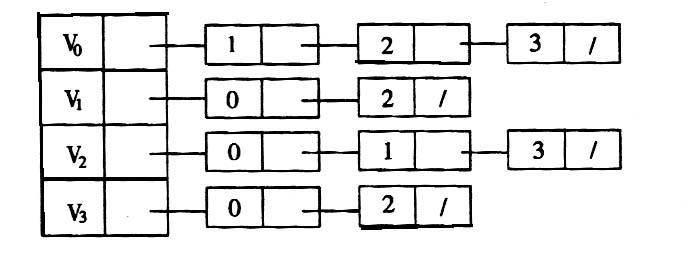
A.由连通网所得到的边数最少的生成树

B.由连通网所得到的顶点数相对较少的生成树

C.连通网中所有生成树中权值之和为最小的生成树

D.连通网的极小连通子图

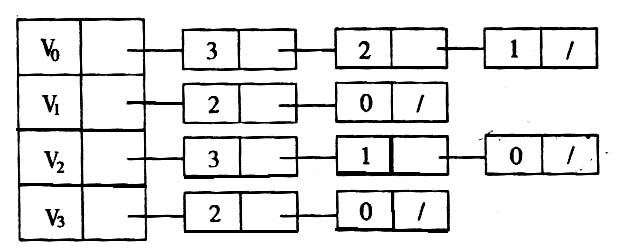
64. 已知图的邻接表如下所示，根据算法，则从顶点V0出发按深度优先遍历的结点序列是（ ）。



A.V0 V1 V3 V2 B.V0 V2 V3 V1

C.V0 V3 V2 V1 D.V0 V1 V2 V3

65. 已知图的邻接表如下所示，根据算法，则从顶点V0出发按广度优先遍历的结点序列是（ ）。



A.V0 V3 V2 V1 B.V0 V1 V2 V3

C.V0 V1 V3 V2 D.V0 V3 V1 V2

66. 迪杰斯特拉（Dijkstra)算法按照路径长度递增的方式求解单源点最短路径问题，该算法运用了（ ）算法策略。

A.贪心 B.分而治之

C.动态规划 D.回溯

67. 下面的排序算法中，最坏情况下计算时间可以达到O（nlogn）的是（ ）。P288

A.直接插入排序 B.希尔排序

C.简单选择排序 D.归并排序

68. 对n个元素的有序表A[1:n]进行顺序查找，其成功查找的平均查找长度（即在查找表中找到指定关键码的元素时，所进行比较的表中元素个数的期望值）为（ ）。P236

A. n B.（n+1)/2 C.log2n D.n2

69. 实现二分查找时，要求查找表（ ）。

A.顺序存储，关键字可以无序排列 B.顺序存储，关键字有序排列

C.双向链表存储，关键字无序排列 D.双向链表存储，关键字有序排列

70. 将5个不同的数据进行直接插入排序，至多需要比较次数为（ ）。   **P276**

**n\*(n-1)/2**

A.8 B.9 C.10 D.25

71. 排序方法中，从未排序序列中依次取出元素与已排序序列（初始时为空）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（ ）。

A.希尔排序 B.冒泡排序 C.插入排序 D.选择排序

72. 从未排序序列中挑选元素，并将其依次插入已排序序列（初始时为空）的一端的方法，称为（ ）。

A.希尔排序 B.归并排序 C.插入排序 D.选择排序

73. 对ｎ个不同的排序码进行冒泡排序，在下列哪种情况下比较的次数最多（ ）。

A.从小到大排列好的 B.从大到小排列好的

C.元素无序 D.元素基本有序

74. 快速排序在下列哪种情况下最易发挥其长处（ ）。

A.被排序的数据中含有多个相同排序码 B.被排序的数据已基本有序

C.被排序的数据完全无序 D. 被排序的数据中的最大值和最小值相差悬殊

75. 对有n个元素的表作快速排序，在最坏情况下，算法的时间复杂度是（ ）。

A.O(n) B.O(n2) C.O(nlog2n) D.O(n3)

76．堆的形状是一棵 （ ）

A.二叉排序树 B.满二叉树 C.完全二叉树 D.类完全二叉树

77. 折半查找有序表（4,6,10,12,20,30,50,70,88,100）。若查找表中元素58，则它将依次与表中（ ）比较大小，查找结果是失败。

A.20,70,30,50 B.30,88,70,50 C.20,50 D.30,88,50

78. 哈希表在查找成功时的平均查找长度（ ）。

A.与处理冲突方法有关，而与装填因子α无关

B.与处理冲突方法无关，而与装填因子α有关

C.与处理冲突方法有关和装填因子α都有关

D.与处理冲突方法无关，也与装填因子α无关

79. 对数据序列{15,9,7,8,20,-1,4}进行排序，进行一趟后数据的排序变为{9,15,7,8,20,-1,4}，则采用的是（ ）算法。

A.简单选择排序 B.冒泡排序 C.直接插入排序 D.堆排序

80. 若对一个链表最常用的操作是在末尾插入结点和删除结点，则采用仅设尾指针的单向循环链表（不含头结点）时，（ ）。

A.插入和删除操作的时间复杂度都为O(1)

B.插入和删除操作的时间复杂度都为O(n)

C.插入操作的时间复杂度都为O(1)，删除操作的时间复杂度都为O(n)

D.插入操作的时间复杂度都为O(n)，删除操作的时间复杂度都为O(1)

81. 下列操作中，( )必须要借助栈来实现。

A.实现函数或过程的递归调用及返回处理时 B.将一个元素序列进行逆置

C.链表结点的申请和释放 D.树的层次遍历算法

82. 以下关于字符串的叙述中，正确的是（ ）。

A.包含任意个空格字符的字符串称为空串

B.字符串不是线性数据结构

C.字符串的长度是指串中所含字符的个数

D.字符串的长度是指串中所含非空格字符的个数

83.若一个栈的入栈序列为“A,B,C,D,E”，则以下不可能的出栈序列是（ ）。 。

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

84．在（ ）中任意一个结点的左、右子树的高度之差的绝对值不超过1。

A.平衡二叉树 B.二叉排序树 C. Huffman树 D.最小生成树

85．如果从无向图的任一顶点出发进行一次深度优先搜索即可访问所有顶点，则该图一定是（ ）。

A. 完全图 B. 连通图 C. 有回路 D. 一棵树

86.以下序列不是堆的是（ ）。

A.{100,85,98,77,80,60,82,40,20,10,66} B.{100,98,85,82,80,77,66,60,40,20,10}

C.{10,20,40,60,66,77,80,82,85,98,100} D.{100,85,40,77,80,60,66,98,82,10,20} 40<60/66

87．在双向循环链表中，在链表中的p节点后插入新节点q，需要执行的操作是（ ）。 。

A．q.next = p.next; q.prev = p; p.next.prev = q; p.next = q;

B. p.next=q；q.prev=p；p.next.prev =q; q.next=q;

C．p.next=q；p.next.prev =q; q.prev =p; q.next =p;

D．q.prev=p；q.next=p.next；p.next=q; p.next.prev=q;

二、填空题

1. 数据结构按逻辑结构可分为两大类，它们分别是 线性结构 和 非线性结构 。

2．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 的关系。

3. 数据的存储结构主要有四种，它们分别是 顺序存储结构、 链式存储结构 、索引存储结构 和 散列存储结构。

4. 向一个长度为n的顺序表的第i个元素（1≤i≤n+1）之前插入一个元素时，需向后移动n-i+1个元素。

5. 在顺序表中访问任意一个元素的时间复杂度均为 O(1)，因此顺序表也称为 随机存取 的数据结构。

6. 用二叉链表法存储包含n个结点的二叉树，结点的2n个指针区域中，有 n+1 个为空指针。P155

7. 算法的设计依赖数据的 逻辑结构，算法的实现依赖数据的 存储结构。

8. 线性表可以采用 顺序 存储结构和 链式 存储结构。

9. 串的逻辑结构是 线性结构；通常采用 顺序 存储结构。

10. 顺序栈用data[0..*n*-1]存储数据，栈顶指针为top，其初始值为0，则元素*x*进栈的操作是 data[++top]=x;。

11.数据结构的概念包含三方面：数据的 逻辑 结构、数据的 存储 结构和对数据的 运算 。

12.数据的逻辑结构主要分为3种： 线性结构，树结构，图结构

13.设单链表中有中间结点 A，front指向A的前一个结点，p 指向新结点 X。在结点 A 的前面插入结点 X 的操作序列为：p.next = front.next；front.next = p; 。

14．在n个结点的单链表中要删除已知结点\*p，需找到它的 前一个结点 ，其时间复杂度为 O(n)。

15．栈是一种特殊的线性表，允许插入和删除运算的一端称为 栈顶。不允许插入和删除运算的一端称为 栈底 。

16．队列 是被限定为只能在表的一端进行插入运算，在表的另一端进行删除运算的线性表。

17. 对于带头结点的链串s，串为空的条件是\_s.head.next == null。

18. 有一个10阶对称矩阵*A*，采用压缩存储方式（以行序为主存储下三角部分，且*A*[0][0]存放在B[1]中），则*A*[8][5]在B中的地址是 B[42]。(1+8)\*8/2+5+1=42

19. 顺序查找含*n*个元素的顺序表，若查找成功，则比较关键字的次数最多为 n 次；若查找不成功，则比较关键字的次数为 n+1 次。

20. 由３个结点所构成的二叉树有 5 种形态。

21. 在顺序循环队列中，front表示队头下标，rear表示队尾下标，队列长度为length；判断队列空的条件为：front == rear; ，判断队列满的条件为 front == (rear+1)%length; 。

22. 中缀表达式A-B\*(C+D/(E-F))对应的后缀表达式 ABCDEF-/+\*-。

23. 将对称矩阵An\*n进行压缩存储，只存储其下三角部分元素，并按照行主序进行线性存储，共需要存储 n\*(n+1)/2 个元素，An\*n的矩阵元素a[i,j](其中0≤i≤j<n)对应的线性存储结构的下标为 i\*(i+1)/2+j。P131

24. 一棵具有3个节点的二叉树有 5 种形态。

25. 设一棵完全二叉树有700个结点，则共有 350 个叶子结点。P153

26. 有向图G用邻接表矩阵存储，其第i行的所有元素之和等于顶点i的 出度。

27. 设有一稀疏图G，则G采用 邻接表 存储较省空间。

28. 设有一稠密图G，则G采用 邻接矩阵 存储较省空间。

30.二分查找法是一种采用 分治 策略的算法，前提条件是：采用顺序存储结构，数据元素 排序 。

31. 设一组初始记录关键字为(72，73，71，23，94，16，5)，则以记录关键字72为基准的一趟快速排序结果为 5，16，71，23，72，94，73。

72，73，71，23，94，16，5

5，73，71，23，94，16，5

5，73，71，23，94，16，73

5，16，71，23，94，16，73

5，16，71，23，94，94，73

5，16，71，23，72，94，73

32. 对于*n*个记录的顺序表进行冒泡排序，在最坏的情况下的时间复杂度是 O(n^2)，若对其进行快速排序，在最坏的情况下的时间复杂度是 O(n^2)。

33. 对含有*n*个元素的数序进行直接插入排序，在最好情况下移动元素的个数是 0，关键字比较的次数是 n-1。P275

34. 用5个权值{3,2,4,5,1}构造的哈夫曼（Huffman）树的带权路径长度是 33 。

15

6 9

3 3 4 5

1 2

3\*2+4\*2+5\*2+1\*3+2\*3=33

35. 设要将序列（Q, H, C, Y, P, A, M, S, R, D, F, X）中的关键码按字母序的升序重新排列，则：

冒泡排序第一趟扫描的结果是 H,C,Q,P,A,M,S,R,D,F,X,Y；

初始步长为4的希尔（shell）排序第一趟的结果是 P,A,C,S,Q,D,F,X,R,H,M,Y ；

快速排序第一趟扫描的结果是 F,H,C,D,P,A,M,Q,R,S,Y,X ；

堆排序初始建堆的结果是 Y,S,X,R,P,C,M,H,Q,D,F,A 。

36. 在堆排序、快速排序和归并排序中，若只从存储空间考虑，则应首先选取 堆排序 方法，其次选取 快速排序 方法，最后选取 归并排序 方法；若只从排序结果的稳定性考虑，应选取 归并排序 ；若只从平均情况下最快考虑，则应选取快速排序方法。

37. 假设在有序线性表a[20]上进行二分查找，则比较1次查找成功的结点数为 1，比较2次查找成功的结点数为2，比较5次查找成功的结点数为5 ，平均查找长度为3.7。

1+2\*2+4\*3+8\*4+5\*5 =74

平均长度是74/20 =3.7

38. 一棵包含10000个叶子节点的完全二叉树中，度为1的节点数为\_1\_ 或\_\_0\_\_。

39. 归并排序算法的时间复杂度为 O(n\*log2n)，空间复杂度为 O(n) 。

三.判断题

1. 链表的物理存储结构具有同链表表达的逻辑结构有一样的顺序。 ( B )

2. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。( B )

3. 在表结构中最常用的是线性表，栈和队列不太常用。( B )

4. 邻接表只能用于有向图的存储，邻接矩阵对于有向图和无向图的存储都适用。( B )

5. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。( A )

6. 栈和链表是两种不同的数据结构。( A )

7. 栈和队列的存储方式既可是顺序方式，也可是链接方式。( A )

8. 快速排序是一种不稳定的排序算法，其最差情况下的时间复杂度为O(nlogn)。( B )

9. 无向图的邻接矩阵是对称的。( A )

10. 逻辑结构相同的数据，只能采用相同的存储方法。( B )

11.健壮的算法不会因非法输入数据而出现莫名其妙的状态。( A )

12.线性表中每个元素都有一个前驱元素和一个后继元素。( B )

13.在单链表中，可以从头节点开始查找任何一个节点。( A )

14. 根据标明空子树的中根遍历序列可以构造出一棵唯一的二叉树。( B )

15.中根遍历一棵二叉排序树输出的关键字是有序的。( A )

16.两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。（ B ）

17.队列是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。（ B ）

18.一个栈的输入序列是12345，则栈的输出序列不可能是12345。（ B ）

19.对平衡二叉树进行中根遍历，可得到结点的有序序列。（ A ）

20.哈夫曼树是带权路径长度最短的树，路径上权值较大的结点离根较近。（ A ）

21.度为2的有序树是二叉树。( B )

22.完全二叉树中的节点可以只有非空的右孩子，而其左孩子为空。 ( B )

23.哈夫曼树可以采用静态三叉链表存储。( A )

24.通过数组的顺序存储结构，按行序优先次序保存了数组的全部数据元素，可以通过寻址公式对数组元素进行随机访问。( A )

25.栈是一种特殊的线性表，对插入元素、删除元素操作的位置做了限制。( A )

26.在HASH表中进行查找运算，根据HASH函数就能确定要查找的元素位置，不需要进行关键字的比较( B )

27.中序遍历一棵二叉排序树，可以得到一个有序的序列。( A )

28.使用一次迪杰斯特拉算法，能求出有向网中任意2个顶点的最短路径。( A )

29.希尔排序算法的时间复杂度为 O(n)。( B )

30.冒泡排序和希尔排序都是不稳定的排序算法。( B )

31.有向图中所有顶点的入度之和等于出度之和。( A )

32.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。（ B ）

33.折半查找的存储结构仅限于顺序存储结构，其元素可以是无序的。（ B ）

34.若二叉树用二叉链表作存贮结构，则在n个结点的二叉树链表中只有n—1个非空指针域。（ A ）

35.二叉树中每个结点的两棵子树的高度差等于1。（ B ）

36.二叉树中每个结点的关键字值大于其左非空子树（若存在的话）所有结点的关键字值，且小于其右非空子树（若存在的话）所有结点的关键字值。（ B ）

37.二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。（ B ）

38.在有向图中，各顶点的入度之和等于各顶点的出度之和。（ A ）

39.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。（ B ）

40.在顺序表中，逻辑上相邻的元素在物理位置上不一定相邻。（ B ）

41.一个无向连通图的生成树是图的极小的连通子图。（ A ）

42.对于一棵非空二叉树，它的根结点作为第一层，则它的第i层上最多能有2i－1个结点。（ B ）

43.当一棵具有n个叶子结点的二叉树的WPL值为最小时，称其树为Huffman树，且其二叉树的形状必是唯一的，因此，Huffman编码一定，达到高频使用的编码更短。（ B ）

44.哈希查找法中解决冲突问题的常用方法是除留余数法。（ B ）

是“线性探测再散列”。

四.简答题

1. 线性表有哪两种存储结构？这两种存储结构相比，分别有哪些优点、缺点？

答：线性表的两种存储结构为：顺序存储结构、链式存储结构

线性存储结构：

优点：存储密度大，存储空间利用率高，可以直接存取数据元素

缺点：插入或删除数据元素时，会引起元素的大量移动，因而降低效率

链式存储结构：

优点：插入或删除元素时很方便，使用灵活

缺点：存储密度小，存储空间利用率低

2. 执行下面的语句时，语句s++的执行次数为多少？

int s=0;

for (i=1;i<n-1;i++)

for (j=n;j>=i;j--)

s++;

n + (n-1) + (n-2) + … + 3

答：n-i+1 i=1 n

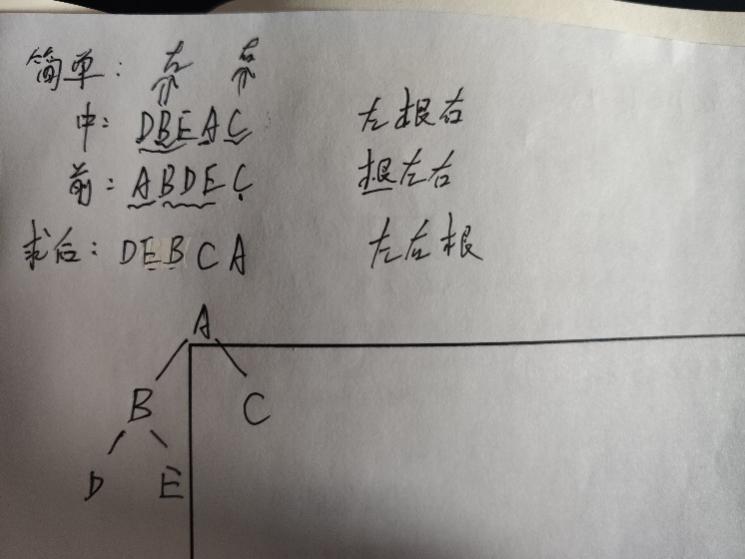
=2 n-1

3 n-2

(n+3)(n-2)/2

3. 设某棵二叉树的中序遍历序列为DBEAC，前序遍历序列为ABDEC，试画出此二叉树，并给出该二叉树的的后序遍历序列。

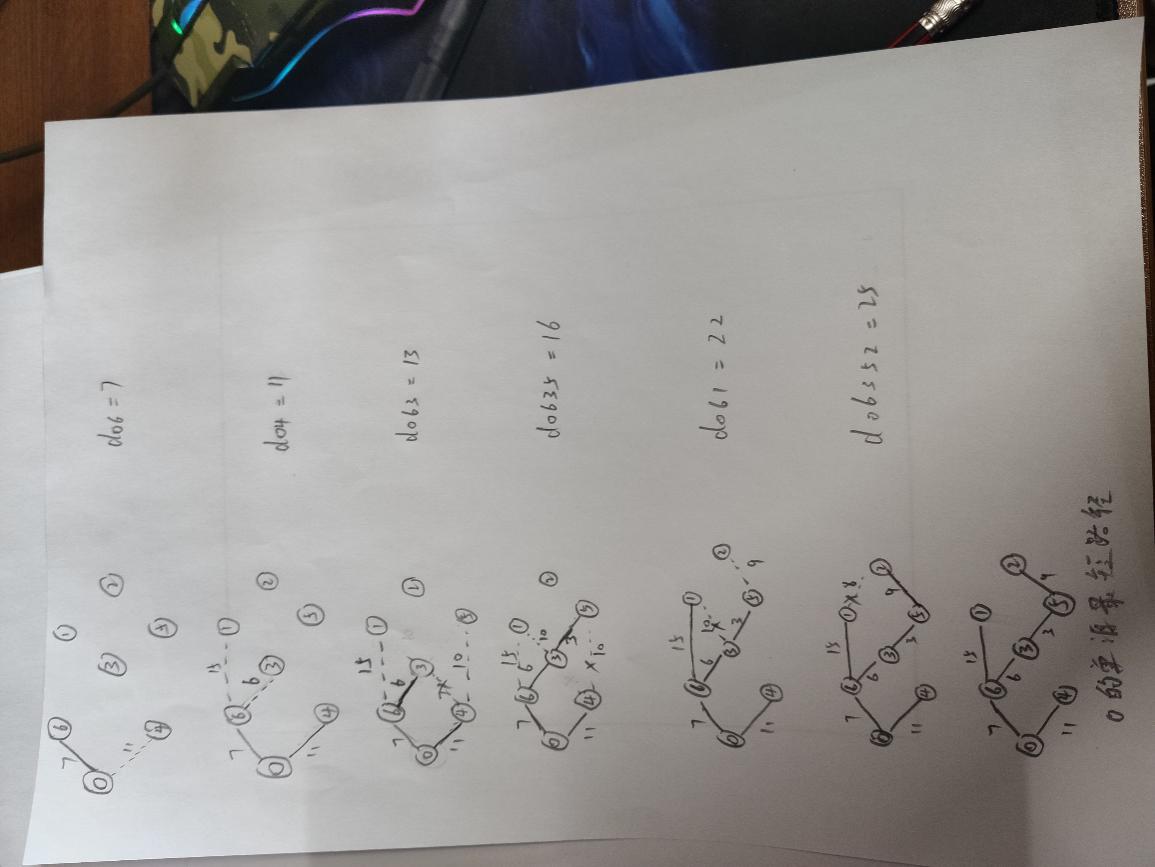
答：DEBCA



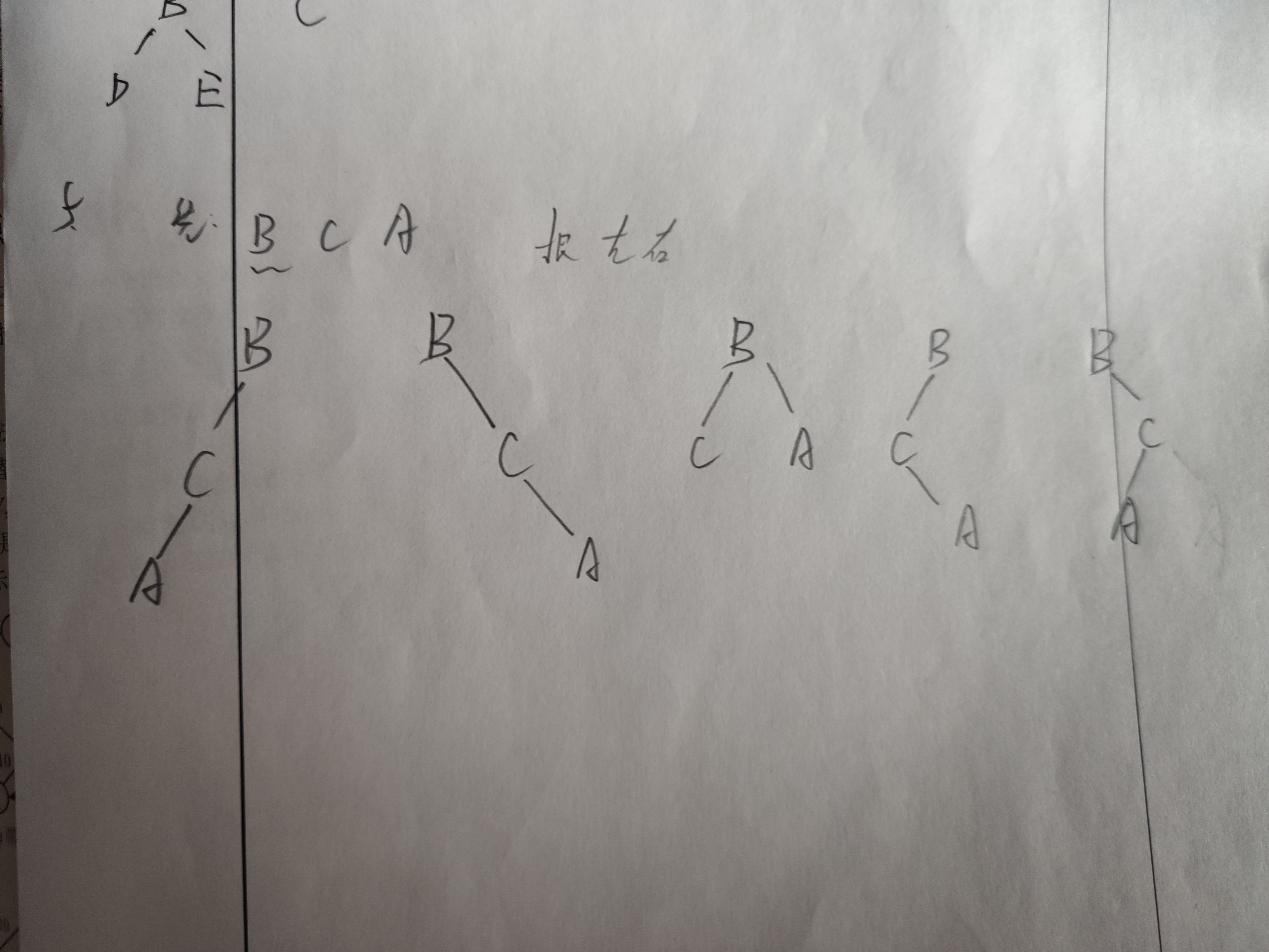
4. 对于如下图所示的带权无向图，求从顶点0到其他各顶点的最短路径。



答：



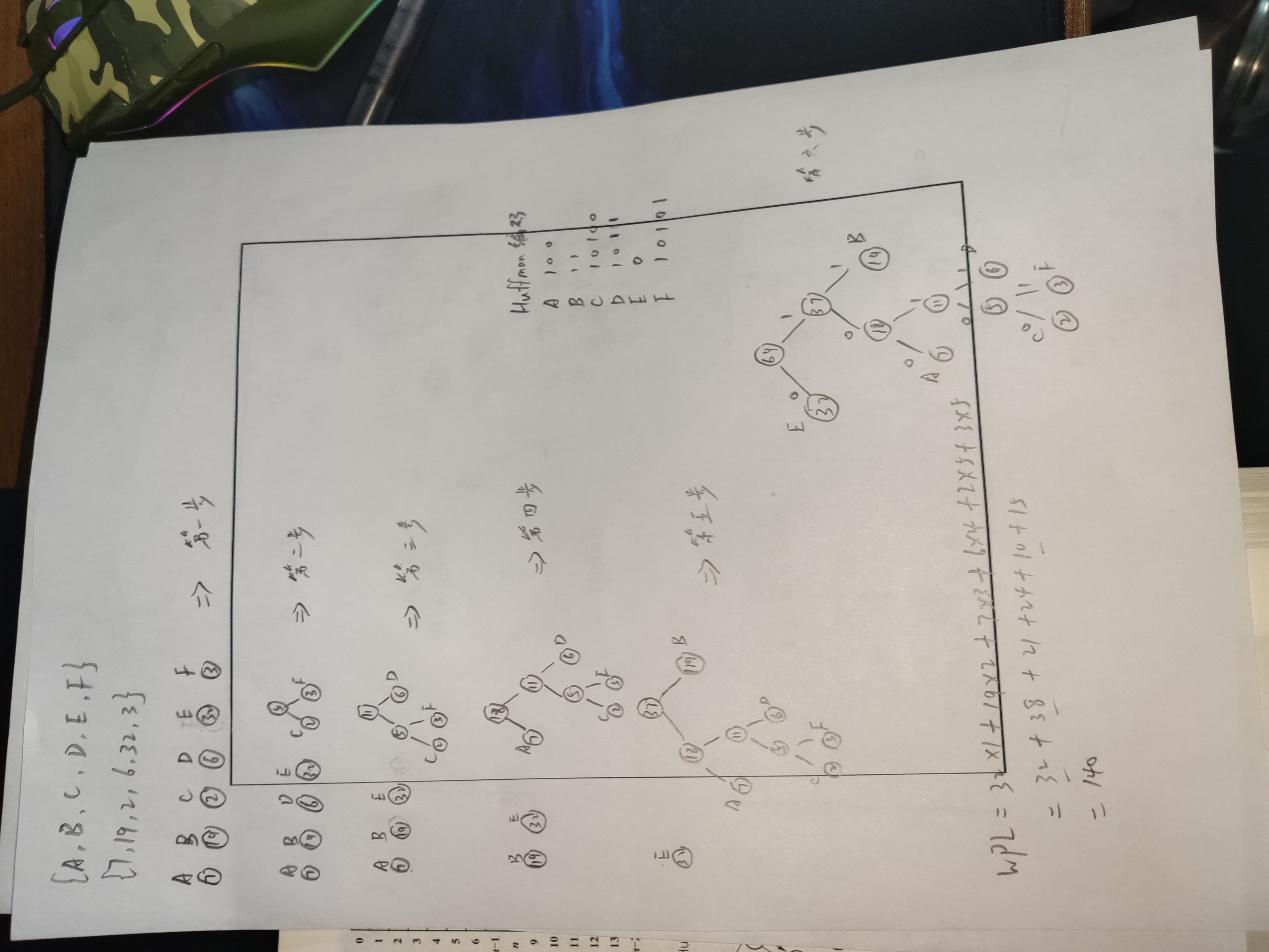
5. 画出先根遍历序列为BCA的所有可能的二叉树形态。



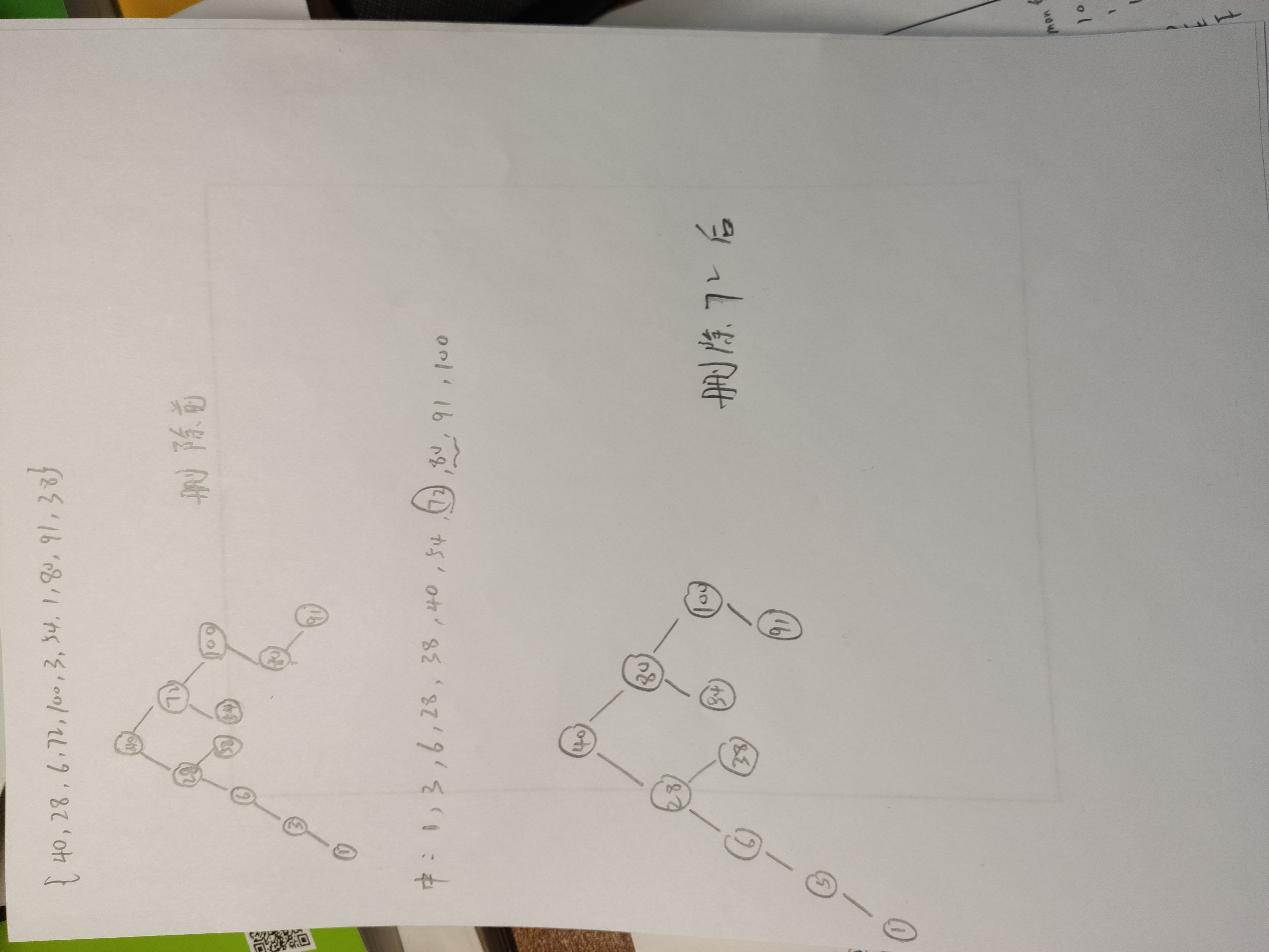
6. 通信过程中的电文仅使用了A~F共6个字母，字母在电文中出现的频率分别为

7, 19, 2, 6, 32, 3。试为这6个字母设计哈夫曼(Huffman)编码，画出相应的哈夫曼树，计算其带权外路径长度WPL，给出及每个字符的哈夫曼编码。（要求哈夫曼树的构造过程中：权值较小的节点为左孩子，权值较大的为右孩子。）

答：



7. 输入一个正整数序列{40,28,6,72,100,3,54,1,80,91,38}，建立一棵二叉排序树，然后删除结点72，分别画出该二叉树及删除结点72后的二叉树。



8. 说明线性表、栈与队列的异同点。

答：

相同点：

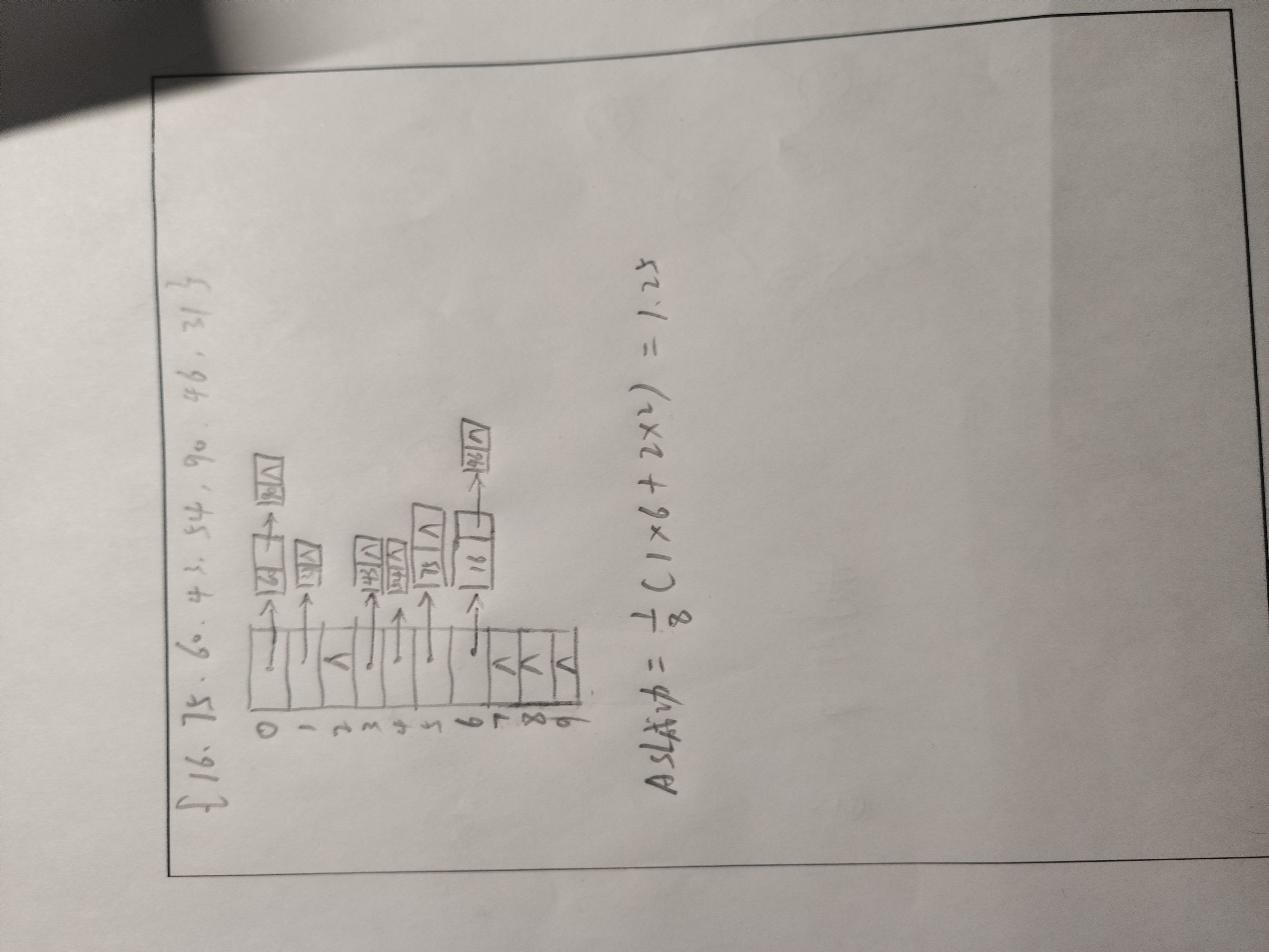
都是线性结构，都是逻辑结构的概念。都可以用顺序存储或链表存储；栈和队列是两种特殊的线性表，即受限的线性表，只是对插入、删除运算加以限制

不同点：

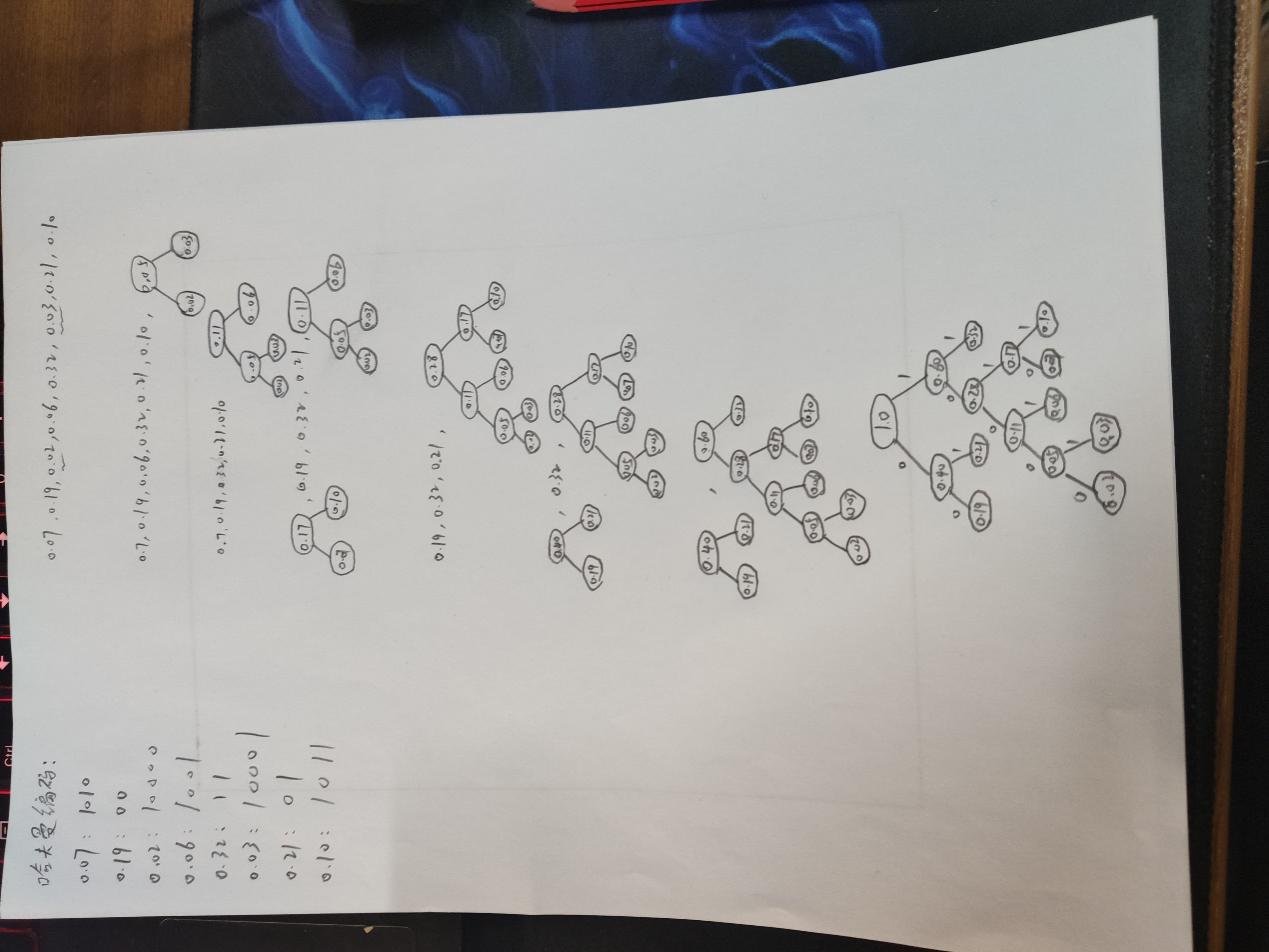
①运算规则不同，线性表为随机存取，而栈是只允许在一端进行插入、删除运算，因而是后进先出表LIFO；队列是只允许在一端进行插入、另一端进行删除运算，因而是先进先出表FIFO。

② 用途不同，堆栈用于子程调用和保护现场，队列用于多道作业处理、指令寄存及其他运算等等。

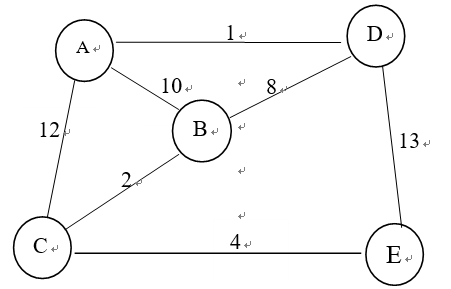
9.对给定的关键字序列{16,75,60,43,54,90,46,31},散列表容量为10，散列函数Hash(key)= key % 10，采用链地址法处理冲突，画出散列表的存储结构图，并计算查找成功的平均查找长度 ASL成功(给出计算表达式)。

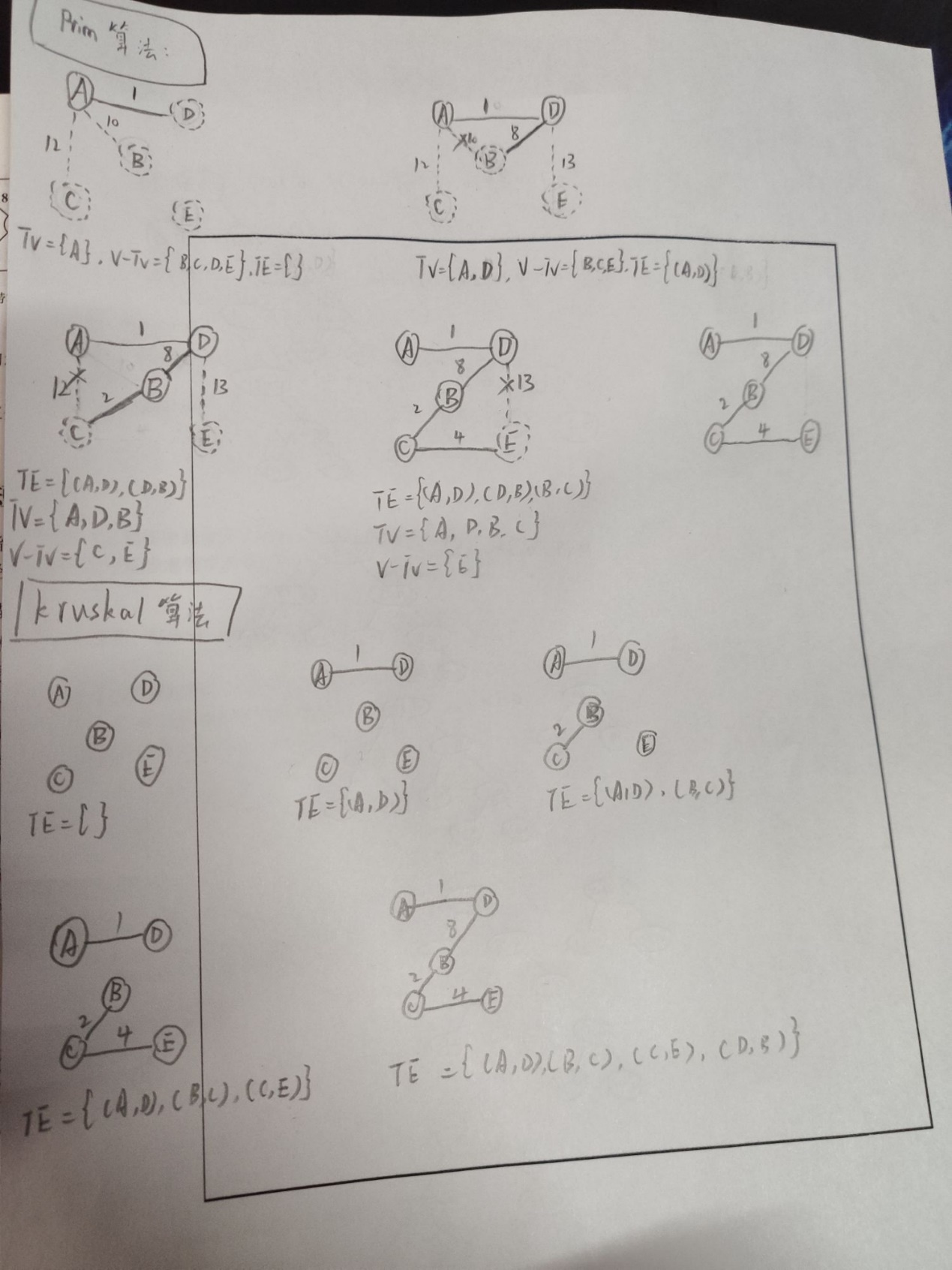


10.假设一组通信的短文由8个字母组成，字母在电文中出现的频率分别为0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.10。试为这8个字母设计哈夫曼编码。若采用等长编码。对于上述实例，比较两种方案的优缺点。



11. 分别采用克鲁斯卡尔(Kruskal)算法和普利姆（prim）算法求解下图的最小生成树，要求给出构造过程。





12. 给出下面有向图的邻接矩阵与邻接表表示，并给出其以A为起点的深度优先遍历。

不唯一P213

5

4

13

7

2

3

9

5

6

A

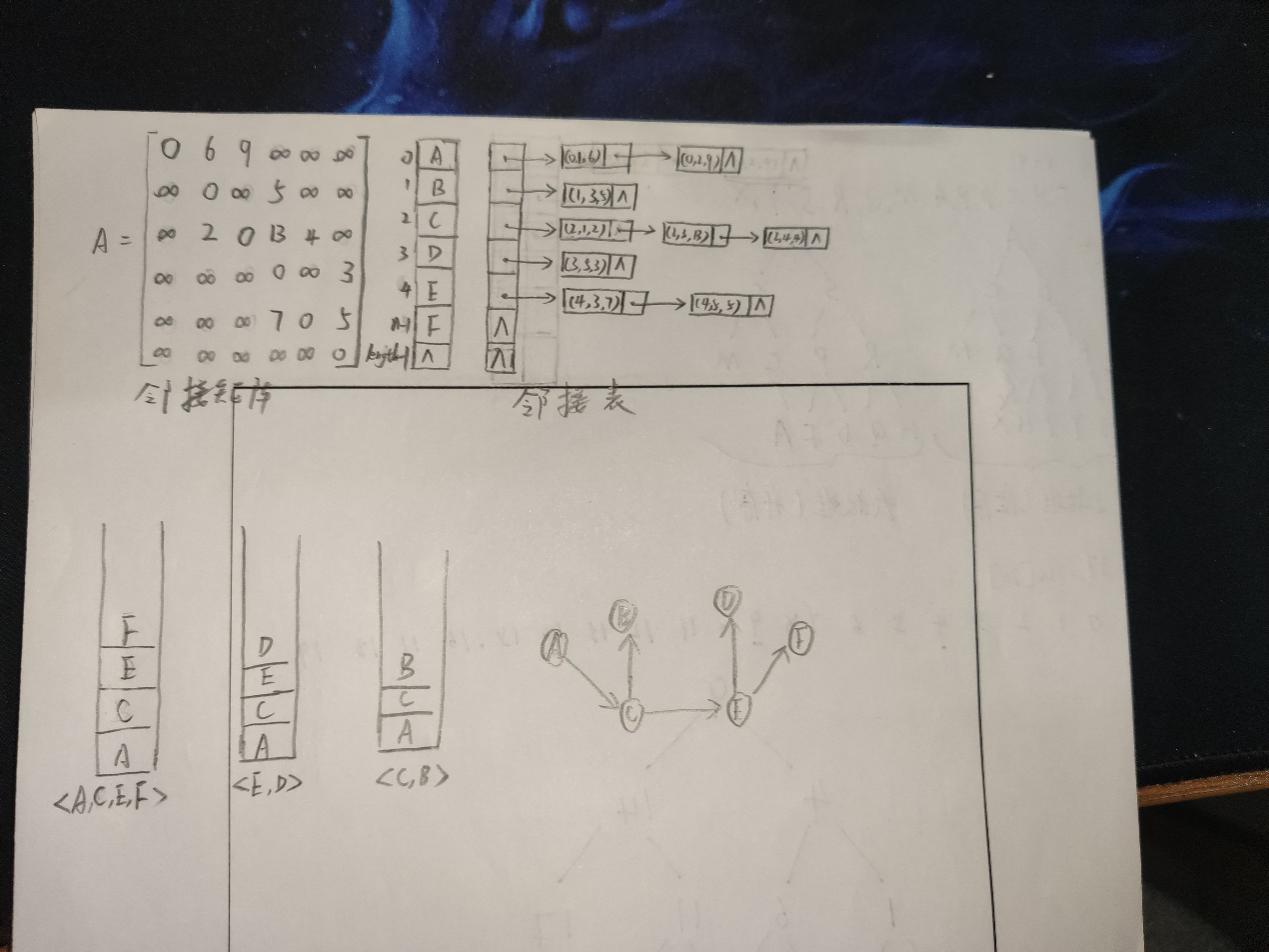
B

D

F

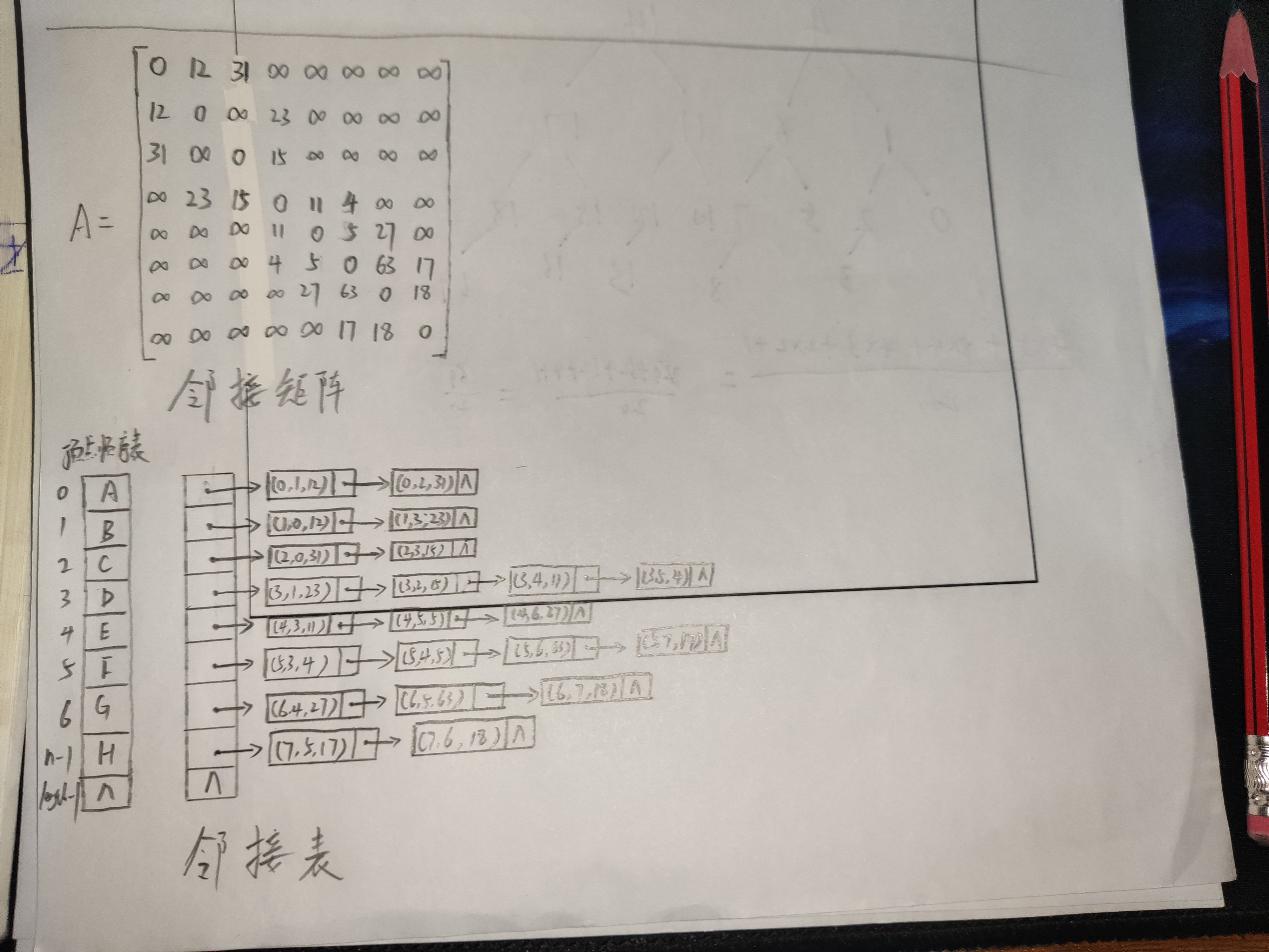
C

E



13. 带权无向图及其约定的顶点次序如图所示，画出邻接矩阵表示和邻接表表示。





14. 设一组初始记录关键字序列为(45，80，48，40，22，78)，降序排序，给出每趟简单选择排序后的结果。选择最大/小的移到前面/后面 本身不变接着下一趟

答：

第1趟：80，45，48，40，22，78

第2趟：80，78，48，40，22，45

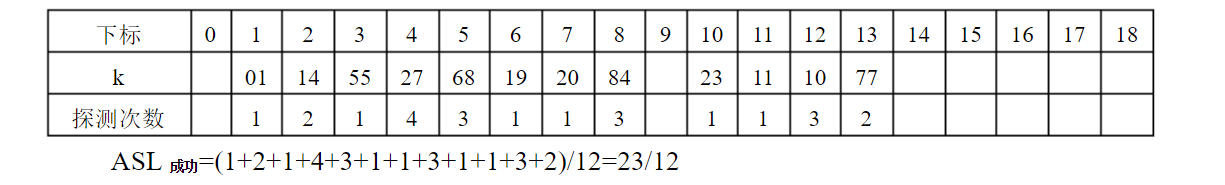
第3趟：80，78，48，40，22，45

第4趟：80，78，48，45，22，40

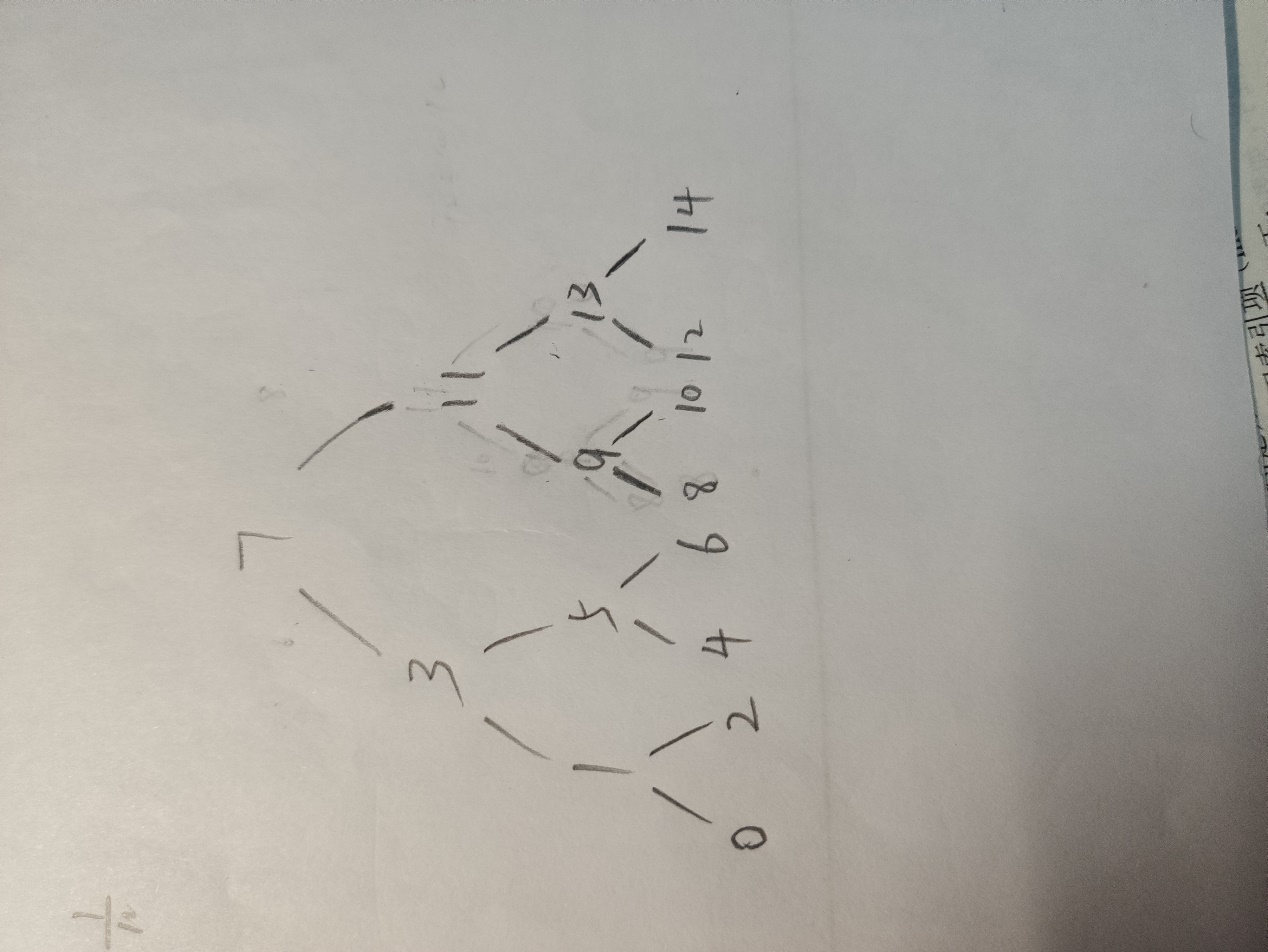
第5趟：80，78，48，45，40，22

15. 设有一组关键字{19,01,23,14,55,20,84,27,68,11,10,77}，采用哈希函数：

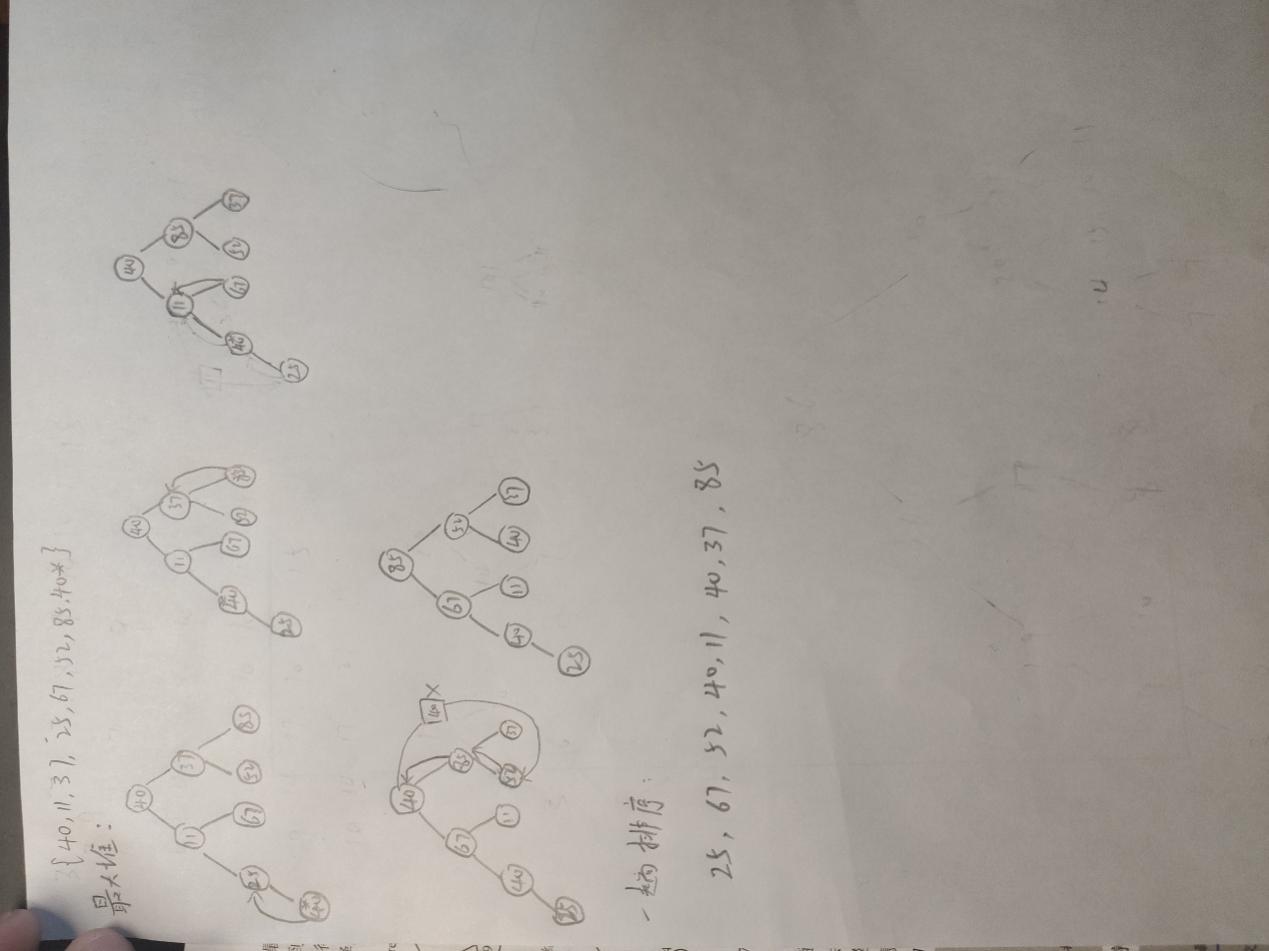
H(key)=key % 13采用开放地址法的线性探测法解决冲突（同义+1），试在0～18的哈希地址空间中对该关键字序列构造哈希表，并求成功情况下的平均查找长度（累加次数/n）。P247

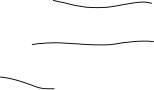
答：

16.有15个关键字按序存储在顺序表中，序号为0，1，2，3，4，5，6，7，8，9，10，11，12，13，14，画出在这个顺序表中进行二分查找时的二叉判定树。

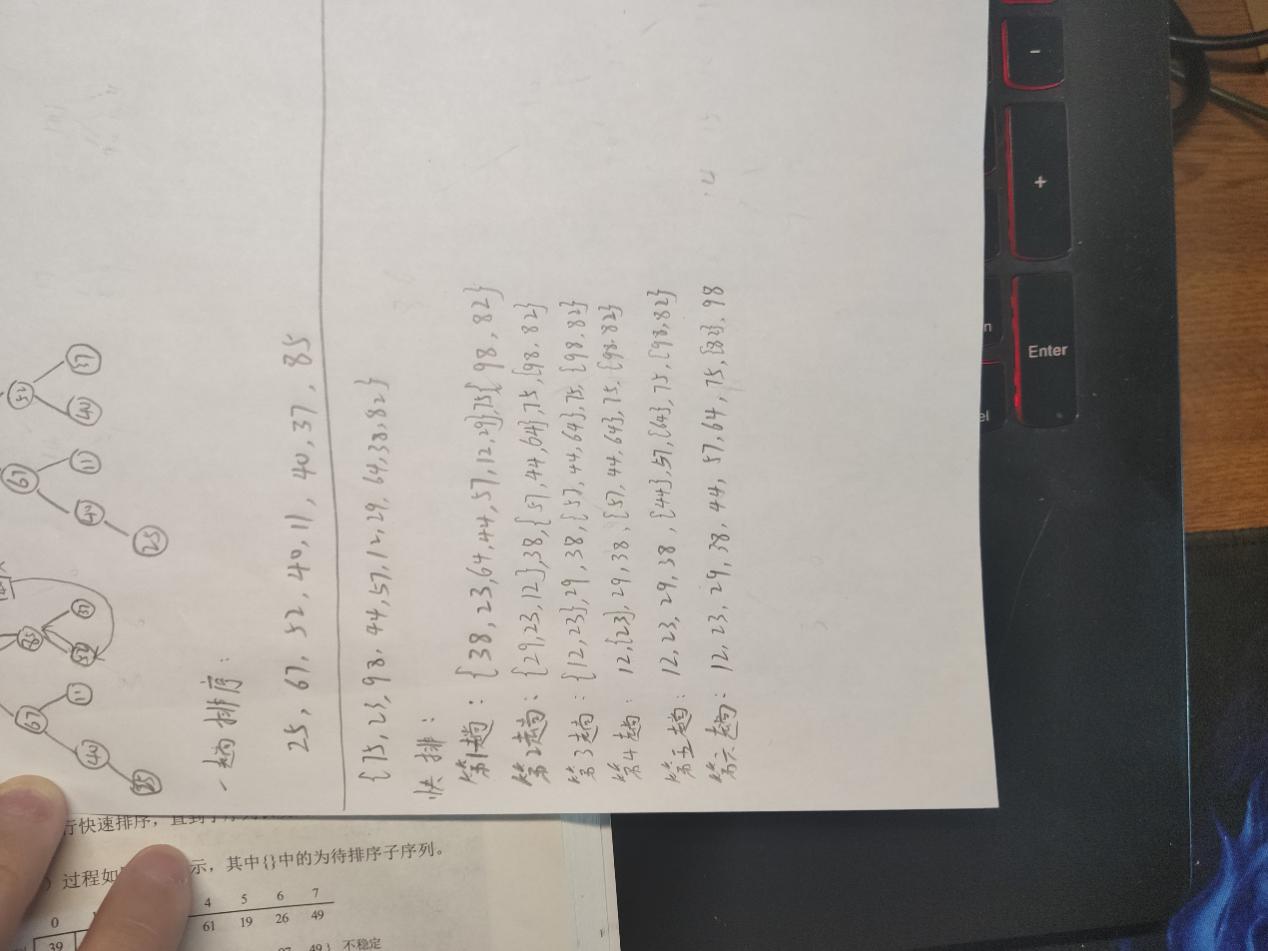


17. 设待排序关键字序列为{40,11,37,25,67,52,85,40\*}，请给出执行一趟堆排序后的结果。**（详细画出堆的形态）**



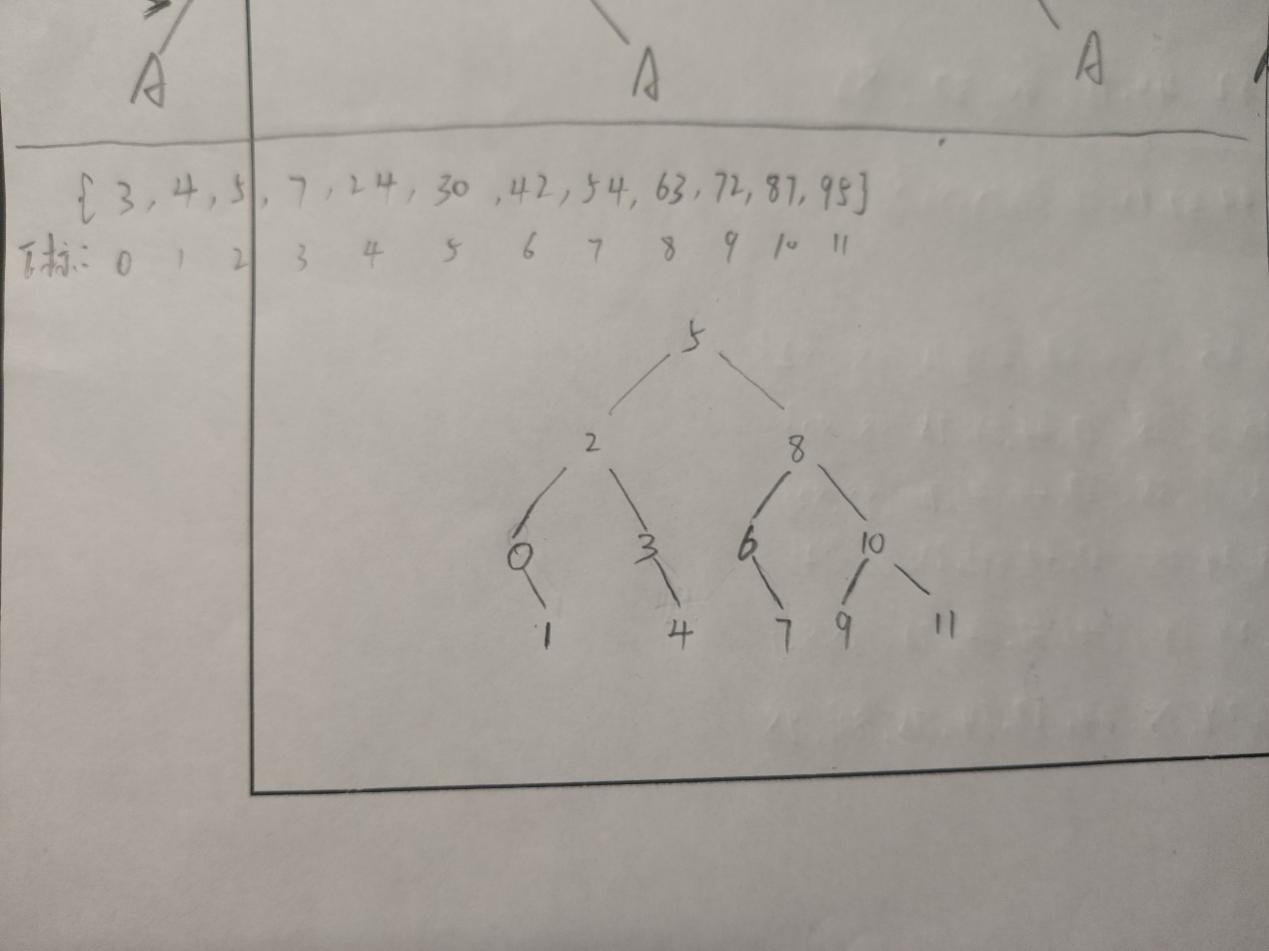


18. 已知序列{75,23,98,44,57,12,29,64,38,82}，给出采用快速排序法对该序列作升序排序时的每一趟的结果。

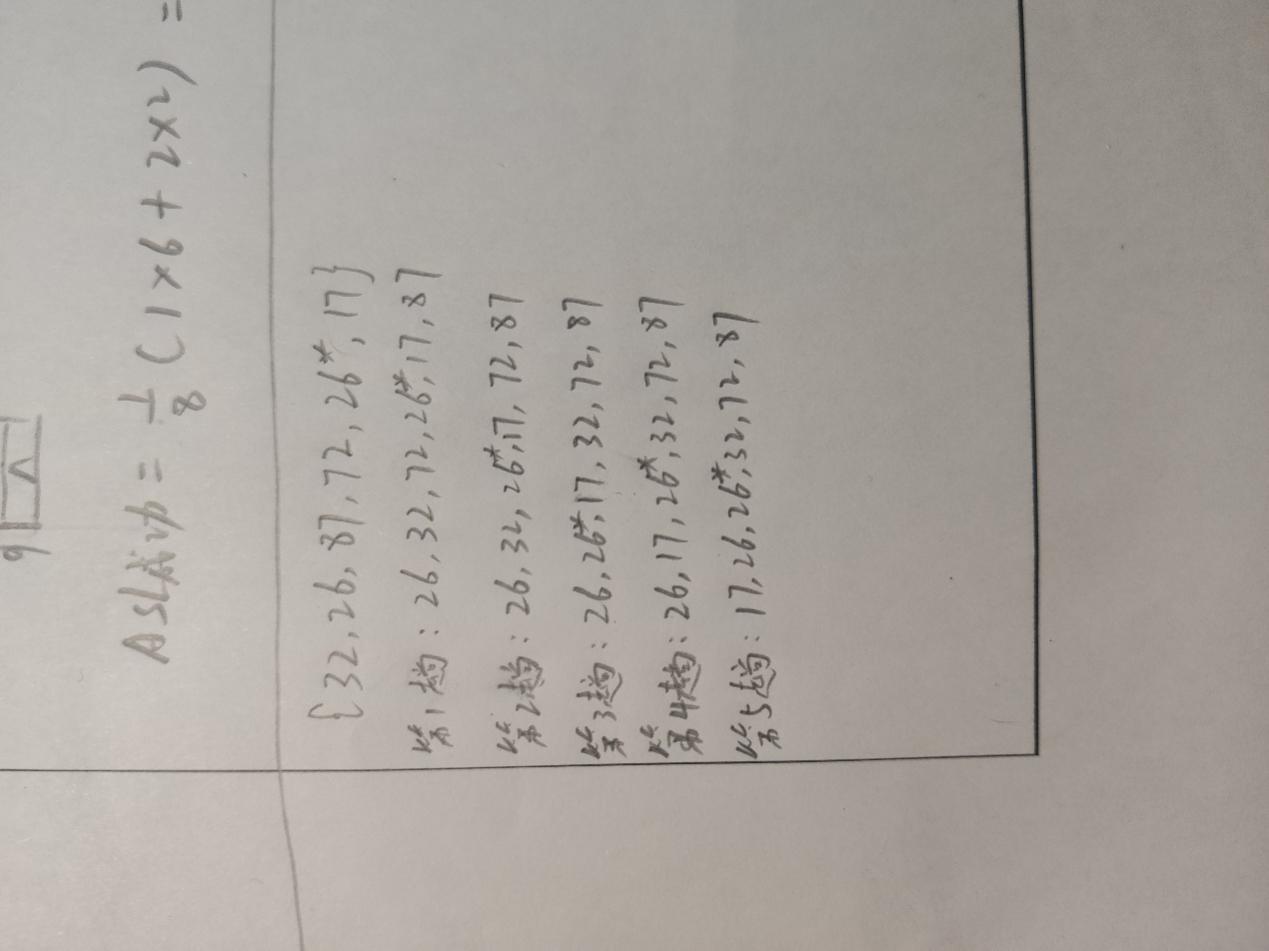




19. 假定对有序表：{3，4，5，7，24，30，42，54，63，72，87，95}进行二分查找，试画出描述二分查找过程的判定树。

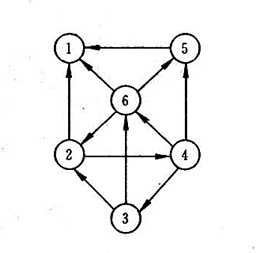


20. 写出关键字序列{32，26，87，72，26\*，17}的冒泡排序（升序）过程。

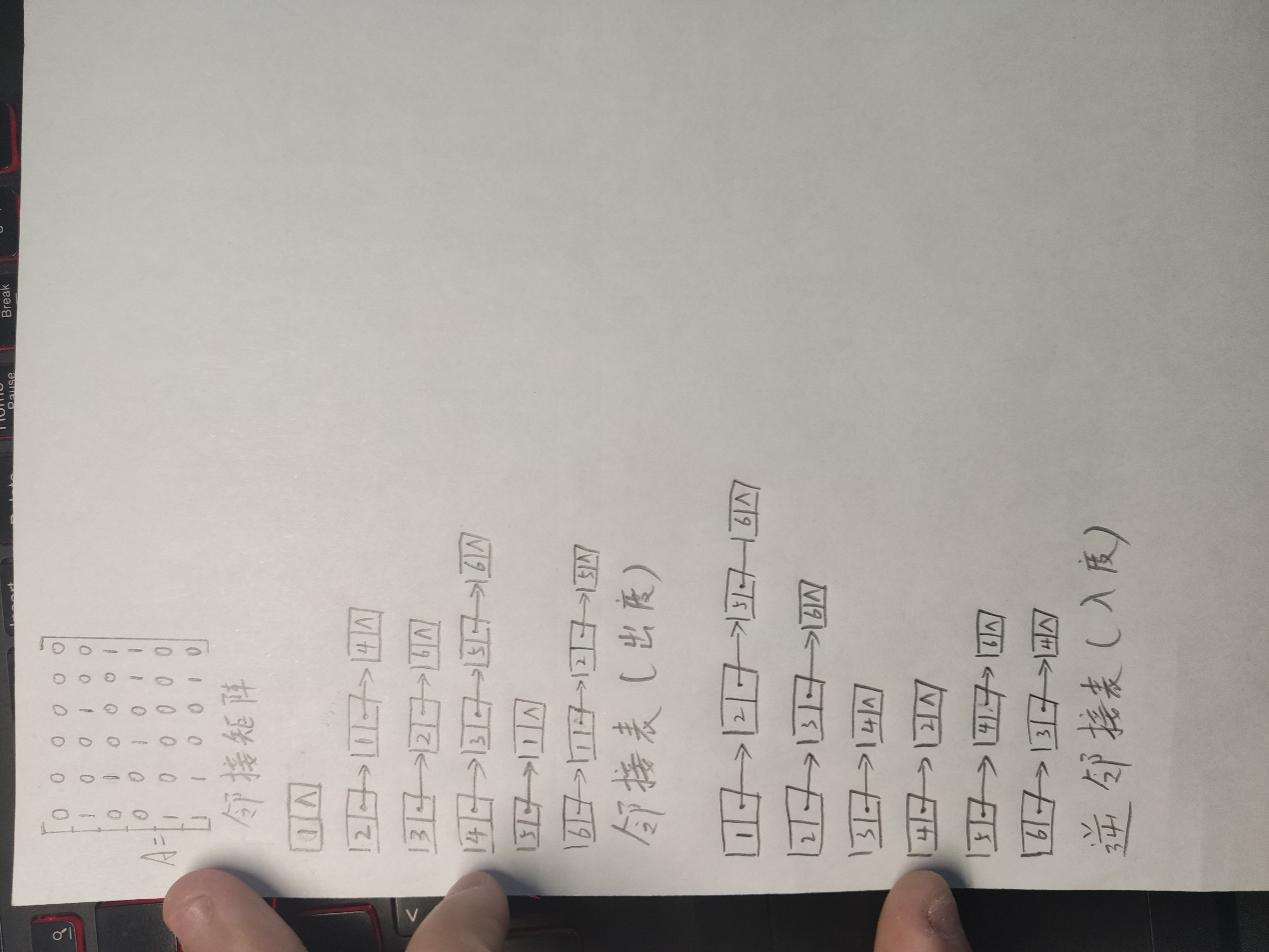


21. 如图所示的有向图，请给出该图的:

1）每个顶点的入/出度；2）邻接矩阵；3）（出边 行）邻接表；4）(入边 列)逆邻接表。



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 顶点 | 1 | 2 | 3 | 4 | 5 | 6 |
| 入度 | 3 | 2 | 1 | 1 | 2 | 2 |
| 出度 | 0 | 2 | 2 | 3 | 1 | 3 |



五 程序设计题

1. 设有一个带头结点的单链表L，编写一个算法删除表中重复的元素。

public T remove(SinglyList<T> L)

    {

        Node<T> p=L.head;

        while(p.next!=null)

        {

            p=p.next;

            if(p.data==p.next.data)

            {

                p.next=p.next.next;

            }

        }

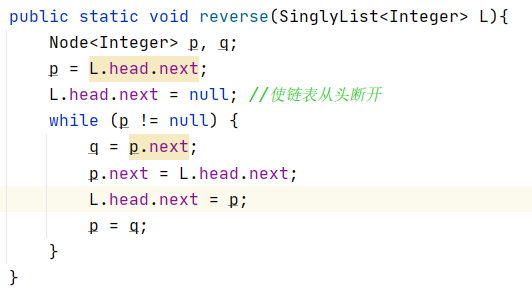
        return null;

    }

2、设有两个带头结点的有序单链表L1和L2，试编写一个算法，将这两个单链表合并成一个单链表，并且合并后单链表也是有序的。



3、设有一个带头结点的单链表L，试编写一个算法实现单链表中数据的逆置。



4、设有一个带头结点的单链表L，试编写删除指定值节点的算法。

    public T remove(T key)

    {

        Node<T> p=this.head;

        while(p.next!=null)

        {

            p=p.next;

            if(p.next.data==key)

            {

p.next=p.next.next;

                return p.next.data;

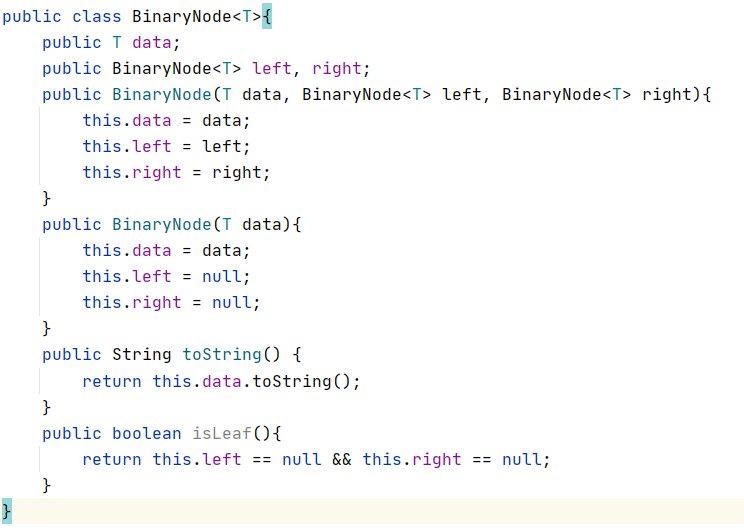
            }

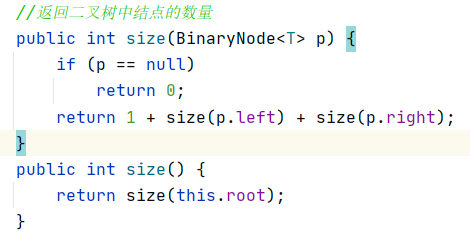
        }

        return null;

    }

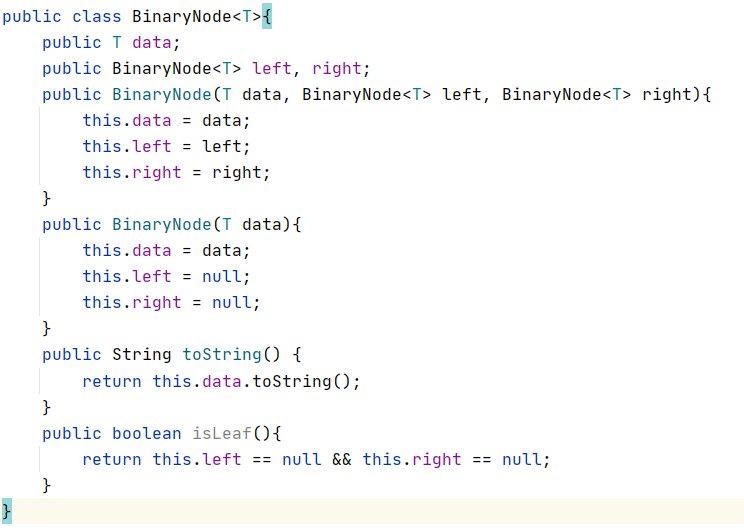
5、给出二叉链表节点类的定义，并编写求解二叉树中节点数目的算法。

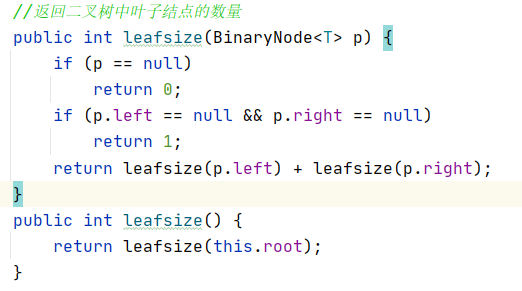




自身+左右子节点

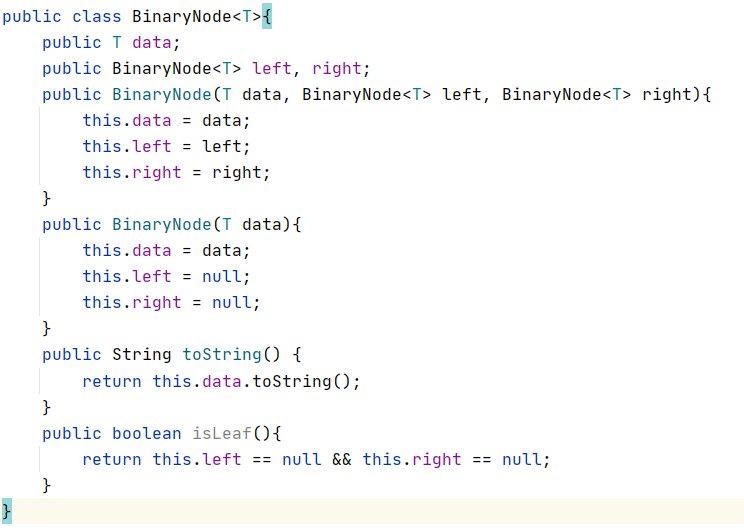
6、给出二叉链表节点类的定义，并编写求解二叉树中叶子节点数目的算法。

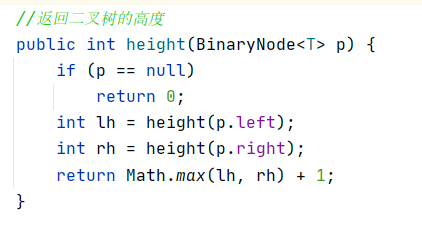




判断是否有左右子节点

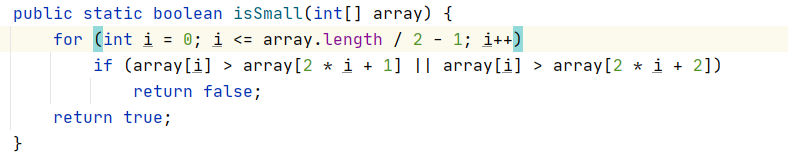
7、给出二叉链表节点类的定义，求二叉树中树的高度的算法。





判断是否有左右子结点 有则+1 取左右最大值返回

8、试编写算法判断一个整数序列是否为小根堆。



判断父节点是否比子节点小 大于则不是小根堆