

# 软工23C1-2023-2024-2《数据结构与算法》模拟测试

开始时间 2024/06/12 14:00:00

结束时间 2024/06/12 14:45:00

答题时长 45分钟

试卷类型 随机试卷

答卷类型 标准答案

试卷总分 100

## 判断题

10 分

### 1-1 随机题目池 1 (5 选 1)

每题 1 分

1-1-1 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用顺序表存储最节省时间。

☒ T

☐ F

| 参考答案

答案 T

1-1-2 对于顺序存储的长度为 $N$ 的线性表，访问结点和增加结点的时间复杂度分别对应为 $O(1)$ 和 $O(N)$ 。

☒ T

☐ F

| 参考答案

答案 T

1-1-3 对于顺序存储的长度为 $N$ 的线性表，删除第一个元素和插入最后一个元素的时间复杂度分别对应为 $O(1)$ 和 $O(N)$ 。

☐ T

☒ F

| 参考答案

答案 F

1-1-4 线性表L如果需要频繁地进行不同下标元素的插入、删除操作，此时选择顺序存储结构更好。

☐ T

☒ F

| 参考答案

答案 F

1-1-5 在具有 $N$ 个结点的单链表中，访问结点和增加结点的时间复杂度分别对应为 $O(1)$ 和 $O(N)$ 。

☐ T

☒ F

| 参考答案

答案 F

## 1-2 随机题目池 2 (3 选 1)

每题 1 分

1-2-1 通过对堆栈S操作: Push(S,1), Push(S,2), Pop(S), Push(S,3), Pop(S), Pop(S)。输出的序列为: 123。

☐ T ☒ F

| 参考答案

答案 F

1-2-2 若一个栈的输入序列为{1, 2, 3, 4, 5}, 则不可能得到{3, 4, 1, 2, 5}这样的出栈序列。

☒ T ☐ F

| 参考答案

答案 T

1-2-3 若一个栈的输入序列为1, 2, 3, ..., N, 输出序列的第一个元素是*i*, 则第*j*个输出元素是*j - i - 1*。

☐ T ☒ F

| 参考答案

答案 F

## 1-3 随机题目池 3 (6 选 1)

每题 1 分

1-3-1 若一个结点是某二叉树的中序遍历序列的最后一个结点, 则它必是该树的前序遍历序列中的最后一个结点。

☐ T ☒ F

| 参考答案

答案 F

1-3-2 若A和B都是一棵二叉树的叶子结点, 则存在这样的二叉树, 其前序遍历序列为...A...B..., 而中序遍历序列为...B...A...。

☐ T ☒ F

| 参考答案

答案 F

1-3-3 某二叉树的前序和中序遍历序列正好一样, 则该二叉树中的任何结点一定都无左孩子。

☒ T ☐ F

| 参考答案

答案 T

1-3-4 已知一棵二叉树的先序遍历结果是ABC, 则CAB不可能是中序遍历结果。

☒ T ☐ F

| 参考答案

答案 T

1-3-5 某二叉树的后序和中序遍历序列正好一样，则该二叉树中的任何结点一定都无右孩子。

☒ T ☐ F

| 参考答案

答案 T

1-3-6 某二叉树的前序和中序遍历序列正好一样，则该二叉树中的任何结点一定都无右孩子。

☐ T ☒ F

| 参考答案

答案 F

1-4 随机题目池 4 (3 选 1)

每题 1 分

1-4-1 给定一棵树，可以找到唯一的一棵二叉树与之对应。

☒ T ☐ F

| 参考答案

答案 T

1-4-2 将一棵树转成二叉树，根结点没有左子树。

☐ T ☒ F

| 参考答案

答案 F

1-4-3 哈夫曼树的结点个数不能是偶数。

☒ T ☐ F

| 参考答案

答案 T

1-5 随机题目池 5 (2 选 1)

每题 1 分

1-5-1 存在一棵总共有2023个结点的二叉树，其中有16个结点只有一个孩子。

☒ T ☐ F

| 参考答案

答案 T

1-5-2 存在一棵总共有2023个结点的二叉树，其中有39个结点只有一个孩子。

☐ T ☒ F

| 参考答案

答案 F

1-6 随机题目池 6 (3 选 1)

每题 1 分

1-6-1 将一棵完全二叉树存于数组中（根结点的下标为1）。则下标为23和24的两个结点是兄弟。

☐ T ☒ F

| 参考答案

答案 F

1-6-2 一棵有124个结点的完全二叉树，其叶结点个数是确定的。

☒ T ☐ F

| 参考答案

答案 T

1-6-3 完全二叉树中，若一个结点没有左孩子，则它必是树叶。

☒ T ☐ F

| 参考答案

答案 T

1-7 随机题目池 7 (3 选 1)

每题 1 分

1-7-1 Kruskal 算法是通过每步添加一条边及其相连的顶点到一棵树，从而逐步生成最小生成树。

☐ T ☒ F

| 参考答案

答案 F

1-7-2 Prim 算法是通过每步添加一条边及其相连的顶点到一棵树，从而逐步生成最小生成树。

☒ T ☐ F

| 参考答案

答案 T

1-7-3 若图G为连通图，则G必有唯一的一棵最小生成树。

☐ T ☒ F

| 参考答案

答案 F

1-8 随机题目池 8 (3 选 1)

每题 1 分

1-8-1 若无向图G必须进行两次广度优先搜索才能访问其所有顶点，则G中一定有回路。

☐ T ☒ F

| 参考答案

答案 F

1-8-2 若无向图G必须进行两次广度优先搜索才能访问其所有顶点，则G一定有2个连通分量。

☒ T ☐ F

| 参考答案

答案 T

1-8-3 图的深度优先遍历非递归算法通常采用栈实现，广度优先遍历非递归算法通常采用队列实现。

☒ T ☐ F

| 参考答案

答案 T

1-9 随机题目池 9 (3 选 1)

每题 1 分

1-9-1 用邻接表法存储图，占用的存储空间数只与图中结点个数有关，而与边数无关。

☐ T ☒ F

| 参考答案

答案 F

1-9-2 无向图中的一条边，在其邻接表存储结构中对应两个弧结点。

☒ T ☐ F

| 参考答案

答案 T

1-9-3 用邻接矩阵法存储图，占用的存储空间数只与图中结点个数有关，而与边数无关。

☒ T ☐ F

| 参考答案

答案 T

1-10 随机题目池 10 (3 选 1)

每题 1 分

1-10-1 对 $N$ 个记录进行快速排序，在最坏的情况下，其时间复杂度是 $O(N\log N)$ 。

☐ T ☒ F

| 参考答案

答案 F

1-10-2 快速排序是稳定的算法。

☐ T

☒ F

| 参考答案

答案 F

1-10-3 希尔排序是稳定的算法。

☐ T

☒ F

| 参考答案

答案 F

单选题

30 分

2-1 随机题目池 1 (5 选 1)

每题 2 分

2-1-1 在数据结构中，从逻辑上可以把数据结构分成（ ）。

- ☐ 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

| 参考答案

答案 C

2-1-2 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

| 参考答案

答案 A

2-1-3 与数据元素本身的形式、内容、相对位置、个数无关的是数据的（ ）。

- ☐ A. 存储结构
- ☐ B. 存储实现
- ☒ C. 逻辑结构
- ☐ D. 运算实现

| 参考答案

答案 C

2-1-4 下列属于线性数据结构的是( )。

- ☒ A. 队列
- ☐ B. 树
- ☐ C. 图
- ☐ D. 二叉树

| 参考答案

答案	A
----	---

2-1-5 以下说法正确的是 ( ) 。

- ☐ A. 数据元素是数据的最小单位
- ☐ B. 数据项是数据的基本单位
- ☐ C. 数据结构是带有结构的各数据项的集合
- ☒ D. 一些表面上很不不同的数据可以有相同的逻辑结构

| 参考答案

答案	D
----	---

2-2 随机题目池 2 (7 选 1)

每题 2 分

2-2-1 对于一个具有 $N$ 个结点的单链表，在给定值为 $x$ 的结点后插入一个新结点的时间复杂度为

- ☐ A.  $O(1)$
- ☐ B.  $O(N/2)$
- ☒ C.  $O(N)$
- ☐ D.  $O(N^2)$

| 参考答案

答案	C
----	---

2-2-2 在单链表中，要删除某一指定结点，必须先找到该结点的 ( ) 。

- ☒ A. 直接前驱
- ☐ B. 自身位置
- ☐ C. 直接后继
- ☐ D. 直接后继的后继

| 参考答案

答案	A
----	---

2-2-3 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. 连续或不连续都可以
- ☐ C. 部分地址必须是连续的
- ☐ D. 一定是不连续的

| 参考答案

答案 B

2-2-4 在具有 $N$ 个结点的单链表中，实现下列哪个操作，其算法的时间复杂度是 $O(N)$ ？

- ☐ A. 在地址为 $p$ 的结点之后插入一个结点
- ☐ B. 删除开始结点
- ☒ C. 遍历链表和求链表的第 $i$ 个结点
- ☐ D. 删除地址为 $p$ 的结点的后继结点

| 参考答案

答案 C

2-2-5 以下关于链式存储结构的叙述中，（）是不正确的。

- ☐ A. 结点除自身信息外还包括指针域，因此存储密度小于顺序存储结构
- ☐ B. 逻辑上相邻的结点物理上不必邻接
- ☒ C. 可以通过计算直接确定第 $i$ 个结点的存储地址
- ☐ D. 插入、删除运算操作方便，不必移动结点

| 参考答案

答案 C

2-2-6 若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用哪种存储方式最节省时间？

- ☐ A. 双链表
- ☐ B. 单循环链表
- ☐ C. 带头结点的双循环链表
- ☒ D. 顺序表

| 参考答案

答案 D

2-2-7 线性表 $L$ 在什么情况下适用于使用链式结构实现？

- ☒ A. 需不断对 $L$ 进行删除插入
- ☐ B. 需经常修改 $L$ 中的结点值
- ☐ C.  $L$ 中含有大量的结点
- ☐ D.  $L$ 中结点结构复杂

| 参考答案

答案 A



2-3-1 某队列允许在其两端进行入队操作，但仅允许在一端进行出队操作。若元素a、b、c、d、e依次入此队列后再进行出队操作，则不可能得到的出队序列是：

- ☐ A. b a c d e
- ☐ B. d b a c e
- ☐ C. e c b a d
- ☒ D. d b c a e

| 参考答案

答案      D

2-3-2 已知初始为空的队列 Q 的一端仅能进行入队操作，另外一端既能进行入队操作又能进行出队操作。若 Q 的入队序列是 1、2、3、4、5，则不能得到的出队序列是：

- ☐ A. 5、4、3、1、2
- ☐ B. 5、3、1、2、4
- ☐ C. 4、2、1、3、5
- ☒ D. 4、1、3、2、5

| 参考答案

答案      D

2-3-3 现有队列 Q 与栈 S，初始时 Q 中的元素依次是{ 1, 2, 3, 4, 5, 6 }（1在队头），S 为空。若允许下列3种操作：  
（1）出队并输出出队元素；（2）出队并将出队元素入栈；（3）出栈并输出出栈元素，则不能得到的输出序列是：

- ☐ A. 1, 2, 5, 6, 4, 3
- ☐ B. 2, 3, 4, 5, 6, 1
- ☒ C. 3, 4, 5, 6, 1, 2
- ☐ D. 6, 5, 4, 3, 2, 1

| 参考答案

答案      C

#### 2-4 随机题目池 4 (3 选 1)

每题 2 分

2-4-1 若元素a、b、c、d、e、f依次进栈，允许进栈、退栈操作交替进行，但不允许连续三次进行退栈工作，则不可能得到的出栈序列是？

- ☐ A. b c a e f d
- ☐ B. c b d a e f
- ☐ C. d c e b f a
- ☒ D. a f e d c b

| 参考答案

答案      D

2-4-2 有六个元素以6、5、4、3、2、1的顺序进栈，问哪个不是合法的出栈序列？

- ☐ A. 2 3 4 1 5 6
- ☒ B. 3 4 6 5 2 1
- ☐ C. 5 4 3 6 1 2
- ☐ D. 4 5 3 1 2 6

| 参考答案

答案      B

2-4-3 将5个字母 `oops` 按此顺序入栈，则有多少种不同的出栈顺序可以仍然得到 `oops`？

- ☐ A. 1
- ☐ B. 3
- ☒ C. 5
- ☐ D. 6

| 参考答案

答案      C

2-5 随机题目池 5 (4 选 1)

每题 2 分

2-5-1 具有1102个结点的完全二叉树一定有\_\_个叶子结点。

- ☐ A. 79
- ☒ B. 551
- ☐ C. 1063
- ☐ D. 不确定

| 参考答案

答案      B

2-5-2 具有65个结点的完全二叉树其深度为（根的深度为1）：

- ☐ A. 8
- ☒ B. 7
- ☐ C. 6
- ☐ D. 5

| 参考答案

答案      B

2-5-3 对一棵满二叉树，m个树叶，n个结点，深度为h，则（）。

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

| 参考答案

答案 D

2-5-4 一个具有1025个结点的二叉树的高 $h$ 为（ ）个。

- ☐ A. 11
- ☐ B. 10
- ☒ C. 11至1025之间
- ☐ D. 10至1024之间

| 参考答案

答案 C

2-6 随机题目池 6 (3 选 1)

每题 2 分

2-6-1 已知字符集{ a, b, c, d, e, f, g, h }。若各字符的哈夫曼编码依次是 0100, 10, 0000, 0101, 001, 011, 11, 0001, 则编码序列 0100011001001011110101 的译码结果是：

- ☐ A. acgabfh
- ☐ B. adbagbb
- ☐ C. afbeagd
- ☒ D. afeefgd

| 参考答案

答案 D

2-6-2 对  $n$  个互不相同的符号进行哈夫曼编码。若生成的哈夫曼树共有 115 个结点，则  $n$  的值是：

- ☐ A. 56
- ☐ B. 57
- ☒ C. 58
- ☐ D. 60

| 参考答案

答案 C

2-6-3 由分别带权为9、2、5、7的四个叶子结点构成一棵哈夫曼树，该树的带权路径长度为：

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

| 参考答案

答案 C

2-7 随机题目池 7 (1 选 1)

每题 2 分

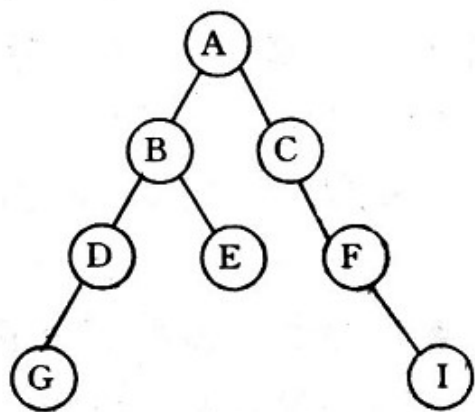
- 2-7-1 若一棵二叉树的后序遍历序列是{ 4, 2, 7, 5, 6, 3, 1 }, 中序遍历序列是{ 4, 2, 1, 5, 7, 3, 6 }, 则下列哪句是错的?
- ☐ A. 这不是一棵完全二叉树
  - ☒ B. 前序遍历序列是{ 1, 2, 4, 3, 7, 5, 6 }
  - ☐ C. 2是4的父结点
  - ☐ D. 设定树根的层次是1, 则这棵树的树高是4

| 参考答案

答案	B
----	---

2-8 随机题目池 8 (4 选 1) 每题 2 分

2-8-1 后序遍历图示二叉树的结果为。

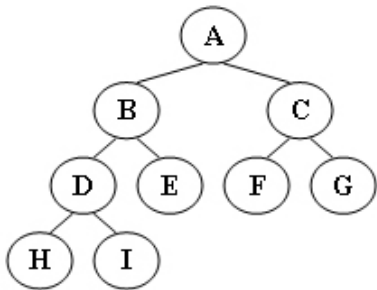


- ☒ A. G, D, E, B, I, F, C, A
- ☐ B. G, D, B, E, I, F, C, A
- ☐ C. G, D, E, B, F, I, C, A
- ☐ D. G, D, B, E, A, C, F, I

| 参考答案

答案	A
----	---

2-8-2 先序遍历图示二叉树的结果为



- ☐ A. A, B, C, D, H, E, I, F, G
- ☒ B. A, B, D, H, I, E, C, F, G
- ☐ C. H, D, I, B, E, A, F, C, G
- ☐ D. H, I, D, B, E, F, G, A, C

| 参考答案

答案 B

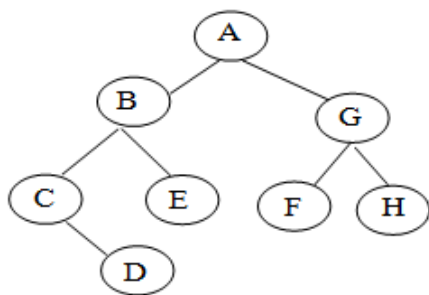
2-8-3 已知一棵二叉树的先序遍历结果是ABC，则以下哪个序列是不可能的中序遍历结果：

- ☐ A. ABC
- ☐ B. BAC
- ☐ C. CBA
- ☒ D. CAB

| 参考答案

答案 D

2-8-4 以下二叉树的后序遍历序列是（ ）。



- ☒ A. DCEBFHGA
- ☐ B. CDBEAFGH
- ☐ C. ABCDEGFH
- ☐ D. ABGCEFHD

| 参考答案

答案 A

2-9 随机题目池 9 (1 选 1)

每题 2 分

2-9-1 由分别带权为9、2、5、2、7、3的6个叶子结点构成一棵哈夫曼树，该树（树根层次为0）的带权路径长度为：

- ☐ A. 75
- ☒ B. 67
- ☐ C. 71
- ☐ D. 63

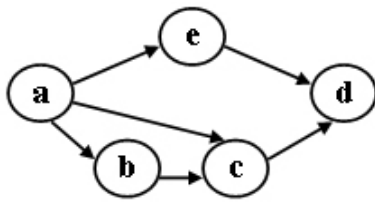
| 参考答案

答案 B

2-10 随机题目池 10 (3 选 1)

每题 2 分

2-10-1 对下图进行拓扑排序，可以得到不同的拓扑序列的个数是：

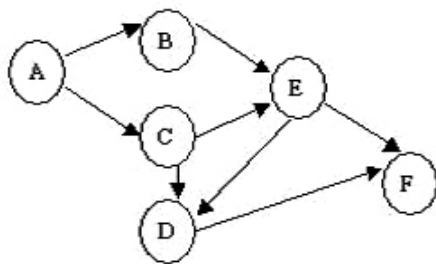


- ☐ A. 4
- ☒ B. 3
- ☐ C. 2
- ☐ D. 1

| 参考答案

答案 B

2-10-2 下图为一个AOV网，其可能的拓扑有序序列为：

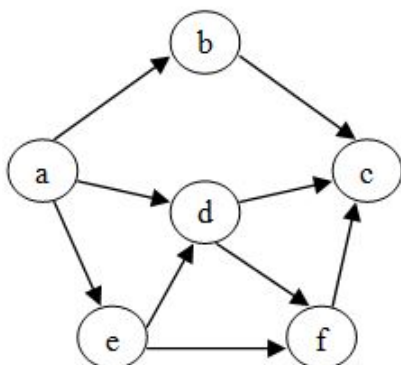


- ☐ A. ACBDEF
- ☐ B. ABCEFD
- ☐ C. ABCDFE
- ☒ D. ABCEDF

| 参考答案

答案 D

2-10-3 给定有向图如下。下列哪个选项不是对应的拓扑序列？



- ☐ A. abedfc
- ☐ B. aedbfc
- ☐ C. aedfbc
- ☒ D. abdfce

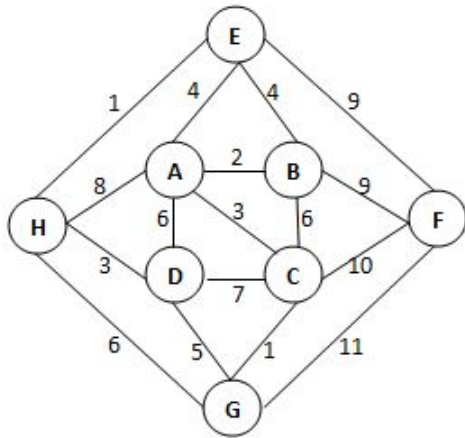
| 参考答案

答案 D

2-11 随机题目池 11 (5 选 1)

每题 2 分

2-11-1 给定有权无向图如下。关于其最小生成树，下列哪句是对的？



- ☒ A. 最小生成树不唯一，其总权重为23
- ☐ B. 最小生成树唯一，其总权重为20
- ☐ C. 边(B, F)一定在树中，树的总权重为23
- ☐ D. 边(H, G)一定在树中，树的总权重为20

| 参考答案

答案 A

2-11-2 给定有权无向图的邻接矩阵如下，其最小生成树的总权重是：

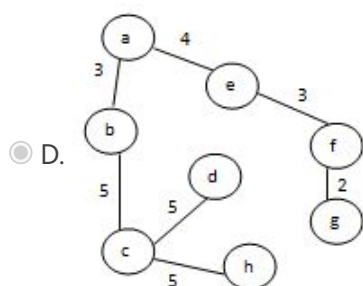
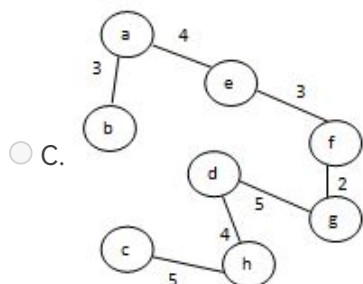
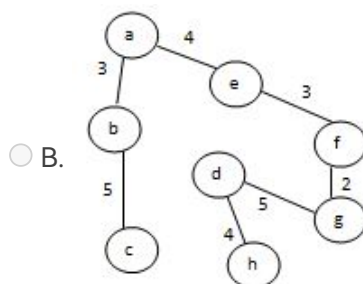
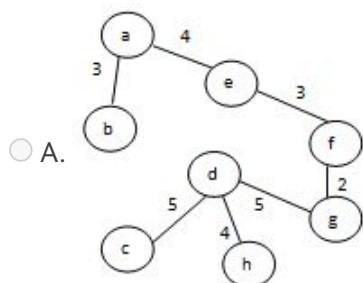
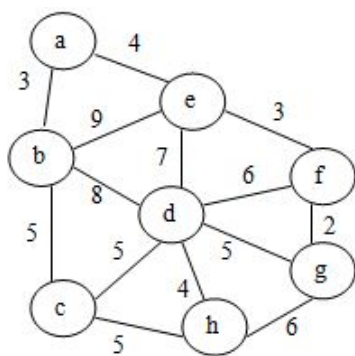
$$\begin{bmatrix} \infty & 3 & 1 & 15 & 1 & 9 \\ 3 & \infty & 3 & 13 & \infty & 8 \\ 1 & 3 & \infty & 1 & 1 & 2 \\ 15 & 13 & 1 & \infty & 18 & \infty \\ 1 & \infty & 1 & 18 & \infty & 2 \\ 9 & 8 & 2 & \infty & 2 & \infty \end{bmatrix}$$

- ☐ A. 20
- ☐ B. 22
- ☒ C. 8
- ☐ D. 15

| 参考答案

答案 C

2-11-3 以下哪个不是给定无向带权图的最小生成树？

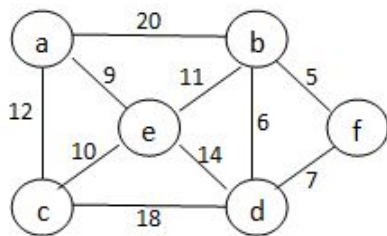


| 参考答案

答案 D

2-11-4 已知无向图  $G$  如下所示，使用克鲁斯卡尔 (Kruskal) 算法求图  $G$  的最小生成树，加入到最小生成树中的边依次是：



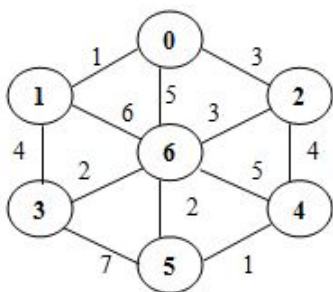


- ☒ A. (b,f), (b,d), (a,e), (c,e), (b,e)
- ☐ B. (b,f), (b,d), (b,e), (a,e), (c,e)
- ☐ C. (a,e), (b,e), (c,e), (b,d), (b,f)
- ☐ D. (a,e), (c,e), (b,e), (b,f), (b,d)

| 参考答案

答案 A

2-11-5 给出如下图所示的具有 7 个结点的网 G，采用Prim算法，从4号结点开始，给出该网的最小生成树。下列哪个选项给出了正确的树结点收集顺序？



- ☐ A. 4501362
- ☐ B. 4526301
- ☐ C. 4561023
- ☒ D. 4563201

| 参考答案

答案 D

2-12 随机题目池 12 (1 选 1)

每题 2 分

2-12-1 在一棵度为3的树T中，若有20个度为3的结点，19个度为2的结点，20个度为1的结点，则树T的叶结点个数是：

- ☐ A. 59
- ☒ B. 60
- ☐ C. 61
- ☐ D. 62

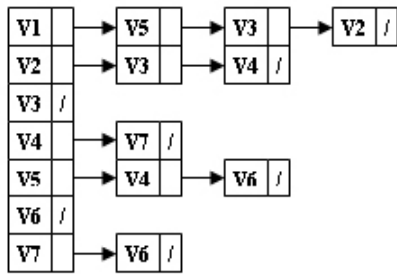
| 参考答案

答案 B

2-13 随机题目池 13 (9 选 1)

每题 2 分

2-13-1 给定一有向图的邻接表如下。从顶点V1出发按深度优先搜索法进行遍历，则得到的一种顶点序列为：

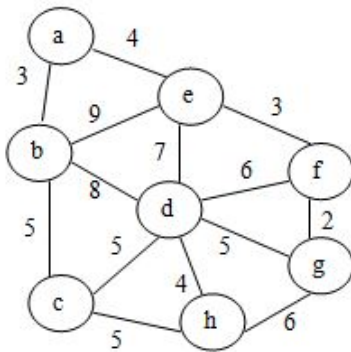


- ☐ A. V1,V5,V4,V7,V6,V2,V3
- ☒ B. V1,V5,V4,V7,V6,V3,V2
- ☐ C. V1,V2,V3,V4,V7,V6,V5
- ☐ D. V1,V5,V6,V4,V7,V2,V3

| 参考答案

答案 B

2-13-2 给定无向带权图如下，以下哪个是从顶点 a 出发深度优先搜索遍历该图的顶点序列（多个顶点可以选择时按字母序）？

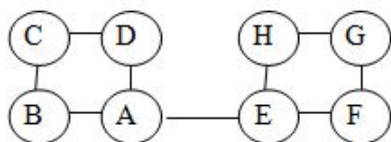


- ☐ A. abecdfhg
- ☐ B. abcdehgf
- ☒ C. abcdefgh
- ☐ D. abchgfde

| 参考答案

答案 C

2-13-3 对下图从顶点C出发进行深度优先搜索，哪个是错误的搜索序列？



- ☐ A. CBADEFGH
- ☒ B. CDABEHFG
- ☐ C. CDAEHGFB
- ☐ D. CBAEFGHD

| 参考答案

答案 B

2-13-4 在用邻接表表示有 $N$ 个结点 $E$ 条边的图时，深度优先遍历算法的时间复杂度为：

- ☐ A.  $O(N)$
- ☒ B.  $O(N + E)$
- ☐ C.  $O(N^2)$
- ☐ D.  $O(N^2 \times E)$

| 参考答案

答案	B
----	---

2-13-5 给定一个图的邻接矩阵如下，则从V1出发的深度优先遍历序列（DFS，有多种选择时小标号优先）是：

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	0	1	1	1	0	0	0	0	0	0
V2	0	0	0	1	1	0	0	0	0	0
V3	0	0	0	1	0	1	0	0	0	0
V4	0	0	0	0	0	1	1	0	1	0
V5	0	0	0	0	0	0	1	0	0	0
V6	0	0	0	0	0	0	0	1	1	0
V7	0	0	0	0	0	0	0	0	1	0
V8	0	0	0	0	0	0	0	0	0	1
V9	0	0	0	0	0	0	0	0	0	1
V10	0	0	0	0	0	0	0	0	0	0

- ☐ A. V1, V2, V4, V3, V6, V8, V10, V9, V7, V5
- ☐ B. V1, V2, V3, V4, V5, V6, V7, V9, V8, V10
- ☒ C. V1, V2, V4, V6, V8, V10, V9, V7, V5, V3
- ☐ D. V1, V2, V3, V5, V7, V9, V10, V6, V8, V4

| 参考答案

答案	C
----	---

2-13-6 下列有关图的叙述中，有几句是对的？

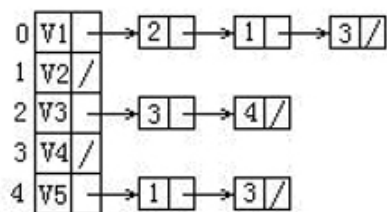
- 1. 如果 $e$ 是有权无向图 $G$ 唯一的一条最短边，那么边 $e$ 一定会在该图的最小生成树上。
- 2. 如果无向图的宽度优先搜索的结果为1234....，且顶点1与顶点4之间存在一条边相连，那么顶点1与顶点3之间也一定有边相连。
- 3. 如果从有向图 $G$ （至少有2个顶点）的每一点均能通过深度优先搜索遍历到所有其它顶点，那么该图一定不存在拓扑序列。
- 4. 若图采用邻接矩阵表示，如果该矩阵不全为0，且矩阵主对角线以下全是0，那么说明该图一定是有向图。

- ☐ A. 3句
- ☒ B. 4句
- ☐ C. 1句
- ☐ D. 2句

| 参考答案

答案	B
----	---

2-13-7 给定一有向图的邻接表如下。从顶点V1出发按广度优先搜索法进行遍历，则得到的一种顶点序列为：

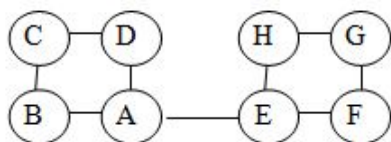


- ☐ A. V1,V2,V3,V4,V5
- ☐ B. V1,V2,V3,V5,V4
- ☒ C. V1,V3,V2,V4,V5
- ☐ D. V1,V4,V3,V5,V2

| 参考答案

答案 C

2-13-8 对下图从顶点C出发进行广度优先搜索，哪个是正确的搜索序列？



- ☐ A. CBDAEFGH
- ☐ B. CDABEHFG
- ☐ C. CBAEHGFD
- ☒ D. CBDAEHFG

| 参考答案

答案 D

2-13-9 给定一个图的邻接矩阵如下，则从V1出发的宽度优先遍历序列（BFS，有多种选择时小标号优先）是：

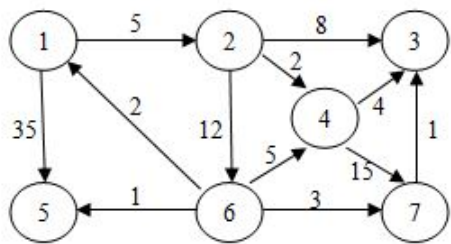
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
V1	0	1	1	1	0	0	0	0	0	0
V2	0	0	0	1	1	0	0	0	0	0
V3	0	0	0	1	0	1	0	0	0	0
V4	0	0	0	0	0	1	1	0	1	0
V5	0	0	0	0	0	0	1	0	0	0
V6	0	0	0	0	0	0	0	1	1	0
V7	0	0	0	0	0	0	0	0	1	0
V8	0	0	0	0	0	0	0	0	0	1
V9	0	0	0	0	0	0	0	0	0	1
V10	0	0	0	0	0	0	0	0	0	0

- ☐ A. V1, V2, V4, V3, V6, V8, V10, V9, V7, V5
- ☒ B. V1, V2, V3, V4, V5, V6, V7, V9, V8, V10
- ☐ C. V1, V2, V4, V6, V8, V10, V9, V7, V5, V3
- ☐ D. V1, V2, V3, V5, V7, V9, V10, V6, V8, V4

| 参考答案

答案 B

2-14-1 使用迪杰斯特拉 (Dijkstra) 算法求下图中从顶点1到其他各顶点的最短路径，依次得到的各最短路径的目标顶点是：

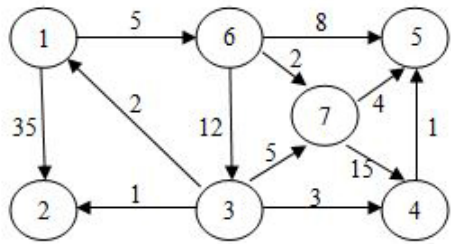


- ☐ A. 2, 5, 3, 4, 6, 7
- ☒ B. 2, 4, 3, 6, 5, 7
- ☐ C. 2, 3, 4, 5, 6, 7
- ☐ D. 5, 2, 6, 3, 4, 7

| 参考答案

答案	B
----	---

2-14-2 使用迪杰斯特拉 (Dijkstra) 算法求下图中从顶点1到其他各顶点的最短路径，依次得到的各最短路径的目标顶点是：

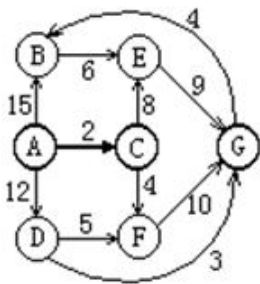


- ☒ A. 6, 7, 5, 3, 2, 4
- ☐ B. 6, 2, 5, 7, 3, 4
- ☐ C. 2, 3, 4, 5, 6, 7
- ☐ D. 2, 4, 3, 6, 5, 7

| 参考答案

答案	A
----	---

2-14-3 试利用 Dijkstra 算法求下图中从顶点 A 到其他顶点的最短距离及对应的路径。下列那个序列给出了可能的顶点收集顺序？



- ☒ A. ACFEDBG
- ☐ B. ACDBFEG
- ☐ C. ACDGFBE
- ☐ D. ABCDEFG

| 参考答案

答案 A

2-15 随机题目池 15 (3 选 1)

每题 2 分

2-15-1 对关键字序列(56, 23, 78, 92, 88, 67, 19, 34), 进行增量为3的一趟希尔排序的结果为 ( )。

- ☐ A. (19, 23, 56, 34, 78, 67, 88, 92)
- ☐ B. (23, 56, 78, 66, 88, 92, 19, 34)
- ☐ C. (19, 23, 34, 56, 67, 78, 88, 92)
- ☒ D. (19, 23, 67, 56, 34, 78, 92, 88)

| 参考答案

答案 D

2-15-2 使用快速排序算法对数据进行升序排序, 若经过一次划分后得到的数据序列是 68, 11, 70, 23, 80, 77, 48, 81, 93, 88, 则该次划分的枢轴是:

- ☐ A. 11
- ☐ B. 70
- ☐ C. 80
- ☒ D. 81

| 参考答案

答案 D

2-15-3 对序列{15, 9, 7, 8, 20, -1, 4} 用希尔排序方法排序, 经一趟后序列变为 {15, -1, 4, 8, 20, 9, 7}, 则该次采用的增量是 ( )。

- ☐ A. 1
- ☒ B. 4
- ☐ C. 3
- ☐ D. 2

| 参考答案

答案 B

函数题

40 分

6-1 随机题目池 1 (1 选 1)

每题 10 分

6-1-1 本题要求实现一个顺序表SeqList的方法, 在顺序表SeqList第*i*个位置 (从1开始) 前插入元素*x*。插入成功返回True, 否则返回False。

假设线性表长度为*n*, 则插入位置*i*须满足 $1 \leq i \leq n + 1$

方法接口定义:

```
1 #在顺序表SeqList第i个位置 (从1开始) 前插入元素x。插入成功返回True, 否则返回False。  
2 def insert(self,i,x):
```

其中顺序表SeqList的类定义如下:

```

1 class SqList:
2     #0. 构造方法
3     def __init__(self):
4         self.initcapacity = 10          #初始容量
5         self.capacity = self.initcapacity #最大存储
6         self.data = [None] * self.capacity #顺序表的数据, 列表
7         self.size = 0                   #顺序表的长度
8     #0. 顺序表的最大容量修改为n
9     def __resize(self,n):
10        assert n >= 0
11        #备份原来的数据
12        a = self.data
13        self.data = [None] * n
14        for i in range(self.size):
15            self.data[i] = a[i]
16    #1. 创建顺序表, 数据源是列表a
17    def create(self,a):
18        for i in range(len(a)):
19            if self.size == self.capacity:#顺序表满了, 2倍扩容
20                self.__resize(self.capacity * 2)
21            self.data[i] = a[i]
22            self.size += 1
23    #2. 输出顺序表
24    def print(self):
25        print("the Length of SqList:",self.size)
26        print("the Elements of SqList:",*self.data[:self.size])
27
28    #3. 你的代码将被嵌在这里, 注意整个方法的代码都要缩进4个空格 (类里面的方法)
29

```

## 裁判测试程序样例:

```

1 sq = SqList()    #创建顺序表
2 a = list(map(int,input().split())) #输入数据到列表a
3 i,x = map(int,input().split())    #输入插入位置i和元素x
4 sq.create(a)        #根据列表a整体创建顺序表sq
5 if sq.insert(i,x):   #位置i前插入元素x成功
6     print("Insert Success")
7     sq.print()
8 else:                #插入失败
9     print("Insert Fail,Index is Error!")

```

## 输入样例1:

输入共有2行, 第1行表示顺序表的元素, 第2行表示插入的位置和元素

```

2 6 4
1 3

```

## 输出样例1:

插入成功, 按样例格式输出。

```

Insert Success
the Length of SqList: 4
the Elements of SqList: 3 2 6 4

```

## 输入样例2:

输入共有2行, 第1行表示顺序表的元素, 第2行表示插入的位置和元素

```
2 6 4
0 8
```

## 输出样例2:

插入失败，按样例格式输出。

```
Insert Fail,Index is Error!
```

## | 参考答案

答案 编译器: PYTHON3

### 6-2 随机题目池 2 (1 选 1)

每题 10 分

6-2-1 本题要求实现一个链表类LinkedList的方法，求链表（含有带头结点）的长度。

#### 方法接口定义:

```
1 #求链表的长度，空表返回0
2 def length(self):
```

其中链表结点类LinkNode和链表类LinkedList的定义如下:

```
class LinkNode:
    def __init__(self,x = None):
        self.data = x          #数据域
        self.next = None       #指针域初始化为空
#单链表类
class LinkedList:
    #0.构造方法
    def __init__(self):
        self.head = LinkNode() #创建一个带头结点
    #1.输出单链表
    def print(self):
        p = self.head.next #p指向链表的第一个有效结点
        if p is None:
            print("空链表")
        else:
            while p.next is not None:
                print(p.data,"->",sep = "",end = "")
                p = p.next
            print(p.data)
    #2.创建单链表-尾插法
    def createTail(self,a):
        tail = self.head #p指向链表的带头结点
        #遍历列表a
        for x in a:
            #根据x创建新结点
            p = LinkNode(x)
            #新结点p链接到链表的尾部
            tail.next = p
            tail = p
    #3.创建单链表-头插法
    def createHead(self,a):
        for x in a:
            p = LinkNode(x)
            p.next = self.head.next
            self.head.next = p
    #4.你的代码将被嵌在这里，注意整个方法的代码都要缩进4个空格（类里面的方法）
```



## 裁判测试程序样例：

```
1 h = LinkedList()    #创建链表
2 a = list(map(int,input().split())) #输入数据到列表a
3 h.createTail(a)      #尾插法创建单链表
4 print(h.length())    #输出链表的长度
```

## 输入样例：

输入共有1行，数据之间用空格分隔

2 0 2 3 0 5 0 4

## 输出样例：

输出链表的长度

8

## | 参考答案

答案

编译器: PYTHON3

### 6-3 随机题目池 3 (1 选 1)

每题 10 分

6-3-1 本题要求计算二叉树中有多少片树叶，输出格式见样例。

## 函数接口定义：

```
#树叶统计
def leafCount(T)
```

其中二叉树类的定义如下：

```
#二叉树的存储-二叉链表
class BinaryTree:
    #1.构造方法
    def __init__(self,newValue):
        self.key = newValue    #树根
        self.left = None       #左子树初始化为空
        self.right = None      #右子树初始化为空
    #2.访问左子树
    def getLeft(self):
        return self.left
    #3.访问右子树
    def getRight(self):
        return self.right
    #4.修改树根的值
    def setRoot(self,newValue):
        self.key = newValue
    #5.访问树根的值
    def getRoot(self):
        return self.key
```

## 裁判测试程序样例：

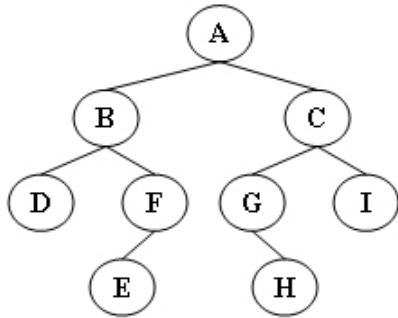
```
T = createBT()    #创建二叉树，实现细节不表
print(leafCount(T))
```

#你的代码将被嵌在这里

输入样例：

```
ABDFECGHI
DBEFAGHCI
```

输出样例（对于图中给出的树）：



4

| 参考答案

答案 编译器: PYTHON3

6-4 随机题目池 4 (1 选 1)

每题 10 分

6-4-1 本题要求实现一个函数，试实现邻接矩阵存储图的广度优先遍历。

函数接口定义：

```
#顶点v(编号)出发对图G进行广度优先遍历
def bfs(G,v)
```

其中图G的定义如下：

```
class adjMatrixGraph:
    # 构造方法，n个顶点m条边
    def __init__(self,n,m):
        self.verNum = n    #顶点数
        self.edgeNum = m    #边数
        self.vertex = [0] * n    #顶点列表
        self.edge = [[0 for i in range(self.verNum)] \
                      for j in range(self.verNum)]    #邻接矩阵二维列表
        self.vis = [False] * n    #顶点的访问列表，默认没访问过
    def addVertex(self,ls): #添加顶点列表
        self.vertex = ls
    def addEdge(self,fr,to):#添加边(fr,to)
        ifr = self.vertex.index(fr)    #起点下标
        ito = self.vertex.index(to)    #终点下标
        self.edge[ifr][ito] = self.edge[ito][ifr] = 1 #邻接矩阵

#邻接矩阵建图
def createGraph():
    n,m = map(int,input().split())    #输入n个顶点和m条边
```

```

g = adjMatrixGraph(n,m)          #创建无向图G
g.addVertex(list(input().split())) #输入顶点列表
for i in range(m):               #输入m条边
    fr,to = input().split()
    g.addEdge(fr,to)
return g                          #返回无向图g

```

#定义抽象类型队列Queue, FIFO (First In,First Out)

```

class Queue:
    #1.构造方法,定义一个空的列表
    def __init__(self):
        self.items = []
    #2.入队,队尾(列表尾部)入队
    def push(self,item):
        self.items.append(item)
    #3.出队,队首(列表头部)出队
    def pop(self):
        return self.items.pop(0)
    #4.判断队列是否为空
    def isEmpty(self):
        return self.items == []
    #5.取队首
    def getFront(self):
        return self.items[0]
    #6.求队列大小
    def getSize(self):
        return len(self.items)

```

#你的代码将被嵌在这里

## 裁判测试程序样例:

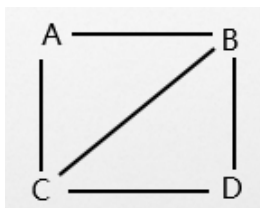
```

g = createGraph()    #创建无向图g
v = int(input())     #输入出发顶点的编号
print("BFS from " + g.vertex[v] + " :",end = "")
bfs(g,v)             #顶点v(编号)出发对图G进行广度优先遍历

```

## 输入样例:

例如无向图



第一行给出图的顶点数n和边数m。第二行给出n个顶点字符串（中间用1个空格分隔），表示n个顶点的数据元素的值。后面是m行，给出每一条边的两个顶点（中间用1个空格分隔）。最后一行输入出发顶点的编号。

```

4 5
A B C D
A B
A C
B C
B D
C D
0

```

## 输出样例:

对于给定的无向图，输出顶点v(编号)出发进行广度优先遍历的顶点（每个顶点前都有1个空格），格式参照样例。

```
BFS from A : A B C D
```

| 参考答案

答案

编译器: PYTHON3

编程题

20 分

7-1 随机题目池 1 (1 选 1)

每题 10 分

7-1-1 给定一串字符，不超过100个字符，可能包括括号、数字、字母、标点符号、空格，编程检查这一串字符中的(),[],{}是否匹配。

输入格式:

输入在一行中给出一行字符串，不超过100个字符，可能包括括号、数字、字母、标点符号、空格。

输出格式:

如果括号配对，输出yes，否则输出no。

输入样例1:

```
sin(10+20)
```

输出样例1:

```
yes
```

输入样例2:

```
{[{}]}
```

输出样例2:

```
no
```

| 参考答案

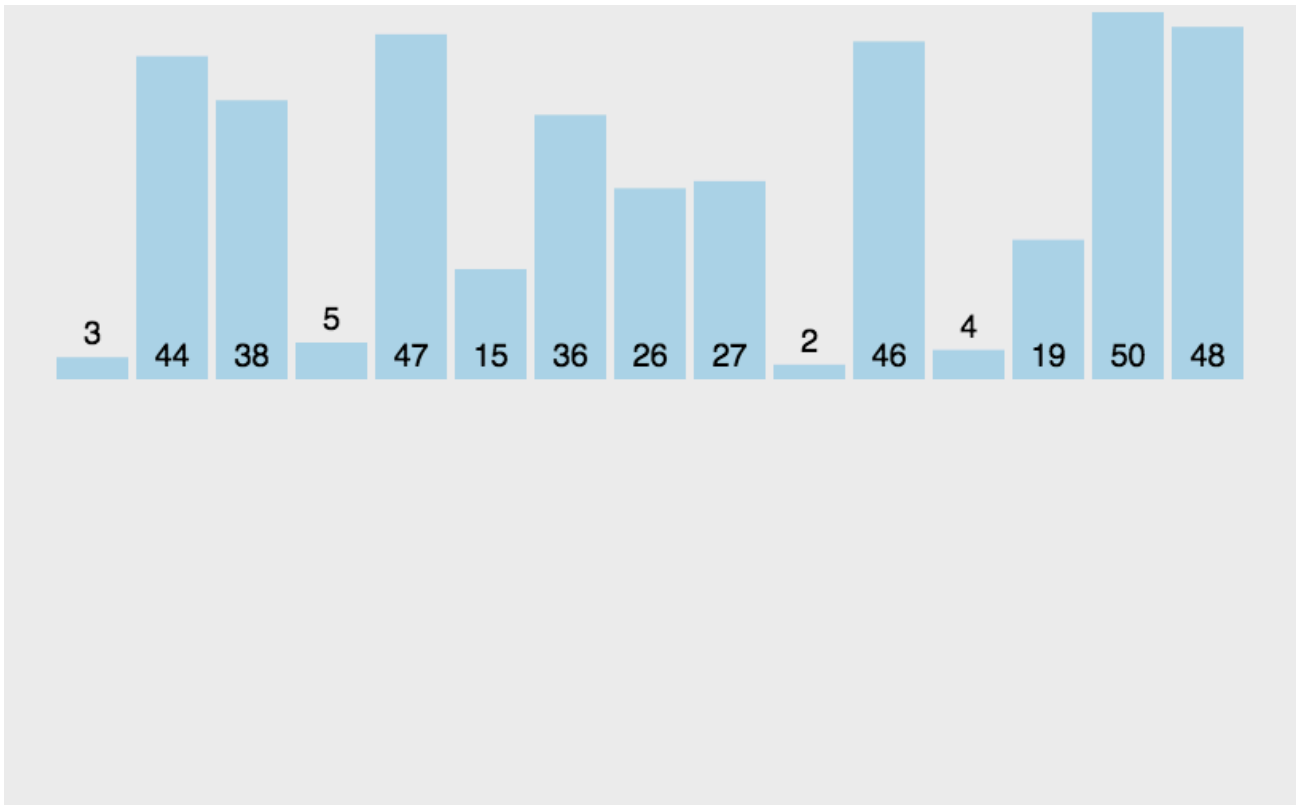
答案

编译器: NO\_COMPILER

7-2 随机题目池 2 (1 选 1)

每题 10 分

7-2-1 **插入排序** 是一种简单直观的排序算法。它的工作原理是通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。



输入格式:

输入在第1行中给出N（ $1 < N \leq 100$ ），在第2行中给出N个待排序的整数，数字间用1个空格分隔。

输出格式:

输出插入排序每一遍后的中间结果序列，数字间用1个空格分隔，但末尾不得有多余空格。

注意，输出时已排序的数据用一对方括号包围，具体格式参见输出样例。

输入样例:

```
7
4 5 7 6 3 2 1
```

输出样例:

```
[4 5] 7 6 3 2 1
[4 5 7] 6 3 2 1
[4 5 6 7] 3 2 1
[3 4 5 6 7] 2 1
[2 3 4 5 6 7] 1
[1 2 3 4 5 6 7]
```

| 参考答案

答案

编译器: NO\_COMPILER