算法提高 插入排序

时间限制：1.0s   内存限制：256.0MB

　　﻿  
　　插入排序

问题描述

　　排序，顾名思义，是将若干个元素按其大小关系排出一个顺序。形式化描述如下：有n个元素a[1]，a[2]，…，a[n]，从小到大排序就是将它们排成一个新顺序a[i[1]]<a[i[2]]<…<a[i[n]]  
　　i[k]为这个新顺序。  
　　插入排序，顾名思义，是通过插入操作完成排序。其直觉和方法来源于打牌时安排牌的方法。每次摸起一张牌，你都会将其插入到现在手牌中它按顺序应在的那个位置。插入排序每一步都类似这个摸牌过程。比如现在有整数数组：{3, 1, 5, 4, 2}  
　　第一步：插入3 得到新序列{3}  
　　第二步：插入1 得到新序列{1 3}  
　　第三步：插入5 得到新序列{1 3 5}  
　　第四步：插入4 得到新序列{1 3 4 5}  
　　第五步：插入2 得到新序列{1 2 3 4 5}  
　　为了看程序中如何完成插入过程，我们以第五步为例：  
　　初始时：1 3 4 5 2  
　　将2存入临时变量tmp  
　　将下标j指向2之前的元素5，然后根据tmp和a[j]的大小关系决定该元素是否应该后移。如果a[j]>tmp，则将a[j]后移到a[j+1]，序列变成1 3 4 5 5。  
　　将下标j前移  
　　判断a[j]>tmp，后移a[j]到a[j+1]，得到1 3 4 4 5  
　　将下标j前移  
　　判断a[j]>tmp，后移a[j]到a[j+1]，得到1 3 3 4 5  
　　因为a[j]<=tmp，所以将tmp放回a[j+1]，得到 1 2 3 4 5  
　　现在，输入n个整数，根据以上算法，输出插入排序的全过程。

输入格式

　　第一行一个正整数n，表示元素个数  
　　第二行为n个整数，以空格隔开

输出格式

　 　有n个元素，因此输出部分分为n个部分，每个部分开头行为：Insert element[i]，i为第几个元素。然后对于每一个部分，输出该部分该元素在插入排序过程中的每一步产生的新序列，初始时的序列以Init:打头，然 后每一步后移数组元素后的元素序列以Move back:打头，最后得到的最终结果序列以Final:打头。序列元素间以一个空格隔开。示例请看样例输出。每一个部分的Insert element[i]之后的每一步的输出行之前要缩进两格，即输出两个空格。

样例输入

5  
3 1 5 4 2

样例输出

Insert element[1]:  
Init:3  
Final:3  
Insert element[2]:  
Init:3 1  
Move back:3 3  
Final:1 3  
Insert element[3]:  
Init:1 3 5  
Final:1 3 5  
Insert element[4]:  
Init:1 3 5 4  
Move back:1 3 5 5  
Final:1 3 4 5  
Insert element[5]:  
Init:1 3 4 5 2  
Move back:1 3 4 5 5  
Move back:1 3 4 4 5  
Move back:1 3 3 4 5  
Final:1 2 3 4 5

数据规模和约定

　　n<=100  
　　整数元素在int范围内

