算法提高 选择排序

时间限制：1.0s   内存限制：256.0MB

　　﻿  
　　选择排序

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。  
　　选择排序的思想极其简单，每一步都把一个最小元素放到前面，如果有多个相等的最小元素，选择排位较考前的放到当前头部。还是那个例子：{3 1 5 4 2}：  
　　第一步将1放到开头（第一个位置），也就是交换3和1，即swap(a[0],a[1])得到{1 3 5 4 2}  
　　第二步将2放到第二个位置，也就是交换3和2，即swap(a[1],a[4])得到{1 2 5 4 3}  
　　第三步将3放到第三个位置，也就是交换5和3，即swap(a[2],a[4])得到{1 2 3 4 5}  
　　第四步将4放到第四个位置，也就是交换4和4，即swap(a[3],a[3])得到{1 2 3 4 5}  
　　第五步将5放到第五个位置，也就是交换5和5，即swap(a[4],a[4])得到{1 2 3 4 5}  
　　输入n个整数，输出选择排序的全过程。  
　　要求使用递归实现。

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

　　共n行，每行输出第n步选择时交换哪两个位置的下标，以及交换得到的序列，格式:  
　　swap(a[i],a[j]):a[0] … a[n-1]  
　　i和j为所交换元素的下标，下标从0开始，最初元素顺序按输入顺序。另外请保证i<=j  
　　a[0]…a[n-1]为交换后的序列，元素间以一个空格隔开

样例输入

5  
4 3 1 1 2

样例输出

swap(a[0], a[2]):1 3 4 1 2  
swap(a[1], a[3]):1 1 4 3 2  
swap(a[2], a[4]):1 1 2 3 4  
swap(a[3], a[3]):1 1 2 3 4  
swap(a[4], a[4]):1 1 2 3 4

数据规模和约定

　　n<=100  
　　整数元素在int范围内

