

# 高级程序设计与实验随堂测试七

- 全局变量被视为不良的编程风格。每声明一个全局变量，相应题目将扣除 10% 的分数。静态 (static) 变量视为全局变量处理。

1. (10 分) 写一个 C++ 函数，输入是一个包含  $n$  个非负整数的向量，并返回一个大小相同的向量。新向量中的第  $i$  个元素应当是输入向量中第  $i$  个元素右侧第一个比它小的元素。如果某个元素右侧不存在更小的元素，则在其对应位置填入  $-1$ 。例如，对于输入  $[4, 3, 27, 25]$ ，输出应该是  $[3, -1, 25, -1]$ 。

函数的时间复杂度必须是  $O(n)$ ，否则最多得一半分。

函数格式: `vector<int> nextLess(const vector<int> & A);`

2. (10分) 写一个C++函数, 它的输入是一个长度为 $n$ 的整型向量参数 $k$  ( $1 < k < n$ ), 打印 (到cout) 当一个宽度为 $k$ 的窗口从向量的起始位置逐位滑动到向量结束位置时, 每个位置下窗口内的最大值。

函数的运行时间必须是 $O(n)$ 的, 否则最多只得一半分。

函数格式: `void window (const vector<int> & A);`

下面是一个简单的测试样例:

```
int main()
{
    window({2, 3, 1, 0, -3, -2, 4, 6, 7}, 3);

    return 0;
}
```

程序输出应是

3, 3, 1, 0, 4, 6, 7

3. (10分) 写一个 C++ 函数, 其输入是一个向量 A 的引用, A 的大小为 n, 保证 A 中的内容都是 1 到 99 之间的整数。函数退出后 A 中的数据应按从小到大顺序排列。

函数运行时间必须是  $O(n)$  的, 否则最多只得一半分。

函数格式: `void smartSort(vector<int>& A);`

4. (Bonus question)

(a, 5 分) 写一个 C++ 函数, 其输入是一个向量 A, A 的内容是 n 个正整数。输出 A 中最长的单调严格递增子序列的长度。(注意, 子序列不需要是连续的。) 若 A 中没有元素, 输出 0。

函数的运行时间应是  $O(n^2)$ 。

函数格式: `int subseq(const vector<int> & A);`

(b, 5 分) 写一个 C++ 函数, 其输入是一个向量 A, A 的内容是 n 个正整数, 这些整数代表的是一个队伍中 n 个同学的身高。输出最小的整数 k, 使得 k 个同学出列后, 剩下的同学形成的队形是一个“合唱队形”。合唱队形的意思是队伍的第一个同学到最高的同学身高严格递增, 从最高的同学到最后一个同学身高严格递减。(最高的同学左边或者右边都可以没有人)

例如: 若输入为 186, 186, 150, 200, 160, 130, 197, 220

输出应是 4, 这是因为将以身高 200 的同学为最高点, 186, 200, 160, 130 可以形成合唱队形, 有四位同学出列。

提示: 本问可以使用(a)问的结果, 即使没做(a)问或没写完, 也可以直接调用, 本问批改时会将 `subseq` 看作能正确完成其功能的函数。

提示: 函数的运行时间应是  $O(n^3)$ 。