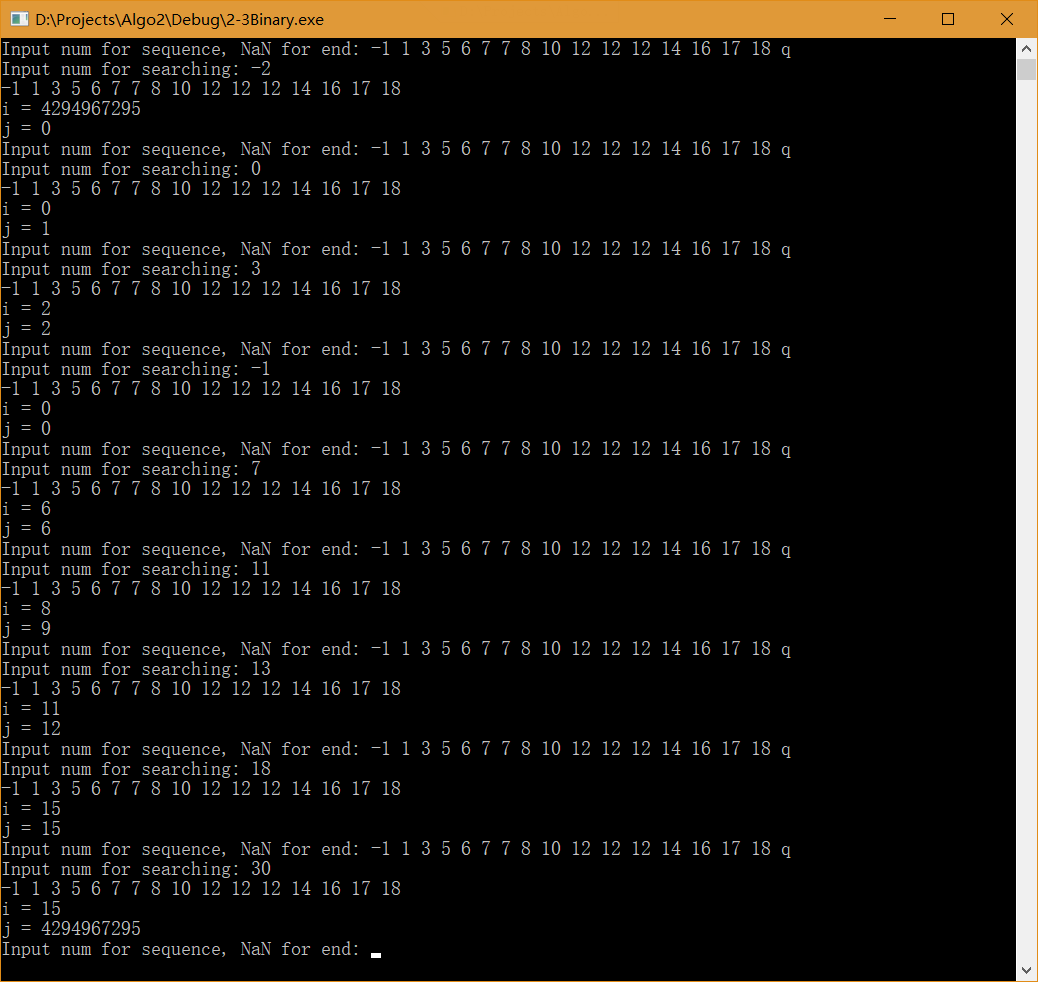
算法设计与分析第二章作业

1652270 冯舜

# 2-3

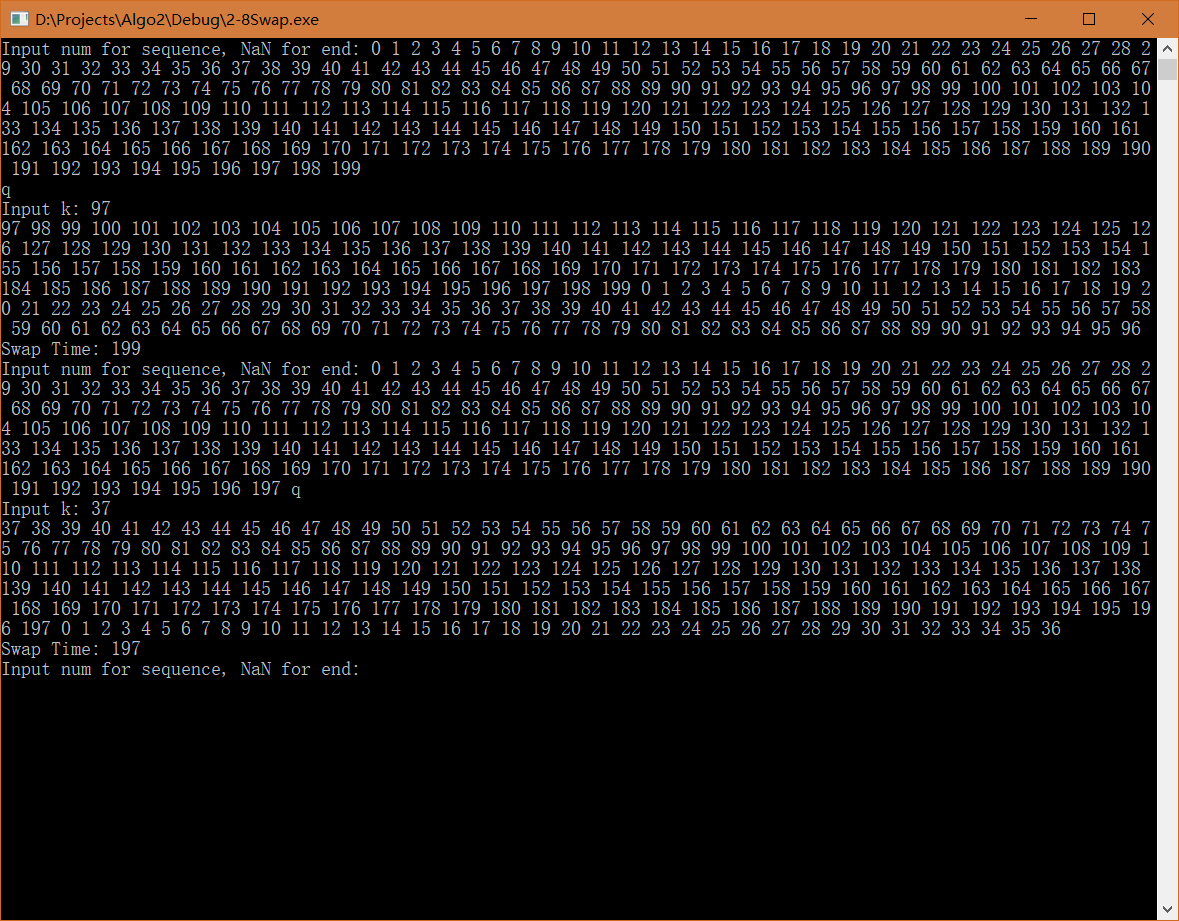
在原二分搜索上所需做的修改为：当搜索的范围start和end相等时，判断该位置值是否为x，是则均将i和j设为该位置返回，否则返回无效值；当搜索的范围start和end只差1时，进行判断其中是否有一个位置的值是x。若是，设置i和j为该位置，返回；否则，判断x应在start左边、end右边或夹在start和end中间，根据情况分别设置i和j（i和j应该只差1）。详见代码2-3.cpp。运行结果如下：

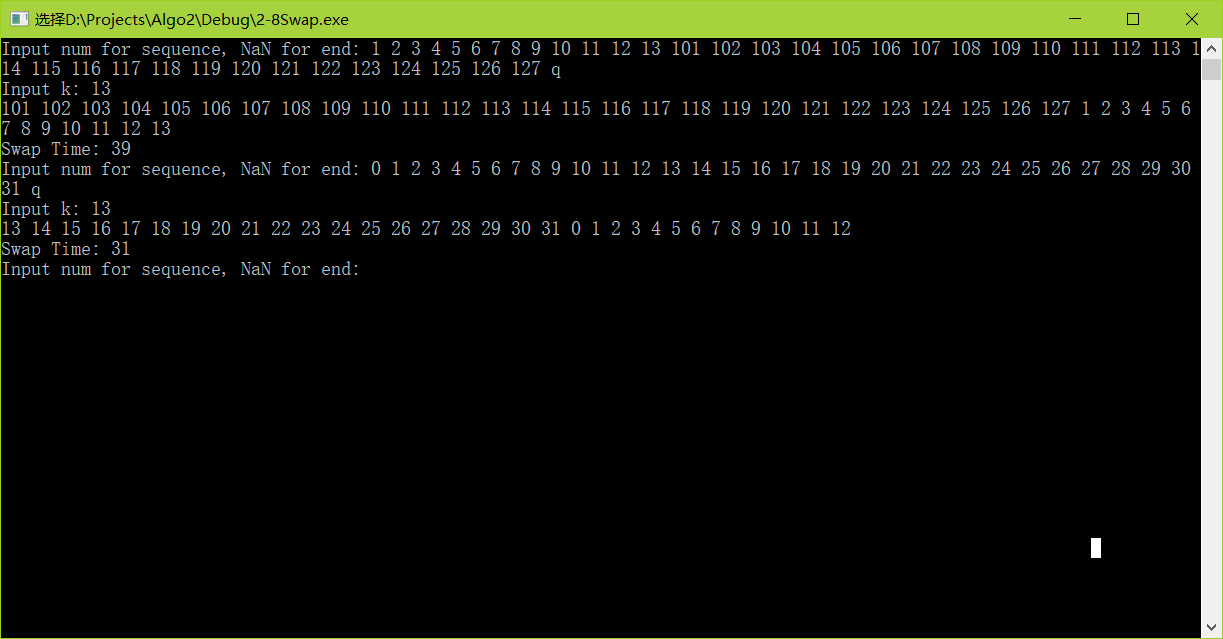


# 2-8

利用分治的思想。若k的设置正好将数组分为长度相等的两数组，显然用n/2次交换即可完成任务；否则，以两数组中较短长度lmin为基准。将原数组分为长度为lmin, l-2lmin, lmin的三部分，k为该三部分其中两数组的边界。先将两侧长度为l的子数组交换位置。这样，只需再将k两侧的两个数组交换位置即可，这是一个更小规模的问题，递归解决。最终，在不超过n次交换的时间复杂度上，可以完成任务。

代码见2-8.cpp。运行结果如下：





# 2-9

此题超出了我的能力范围，故暂不作答。

# 算法实现题2-1

用期望时间复杂度O(n)的求中位数算法，求该集合的中位数M。并用一个函数partition将存储该集合的无序列表按序排为 比M小，等于M，大于M三部分。这样便知M的重数。若比M小的部分的长度大于等于M的重数，则递归求比M小的部分的众数和重数；若比M大的部分的长度大于等于M的重数，则递归求比M大的部分的众数和重数。（该递归终会终结并回代，因为在个数很小时不满足继续递归的条件）比较这三个（或两个、一个）重数，大者所对应的数便作为众数，完成任务。

时间复杂度为O(nlgn)。

详细代码见2-i1.cpp。运行结果如下：

