

## EX1

王嵘晟 PB1711614

### EX-1

思路：此题首先使用结构体数组来存储输入的数值以便于接下来的排序和查找。由于数据量比较大于是排序采用归并排序，按照 $a_{id}$ 的值从小到大排序。由于每次输入FIND要返回一个查找的值，所以查找算法的时间复杂度应该要小于 $O(n)$ ，因此使用二分查找。建立一个flag标记，当且仅当第一次输入FIND时做排序。对于INSERT和FIND的输入，巧妙采用只读取字符串数组的首字母来判断命令。

由于归并排序的时间复杂度为 $O(n\log_2 n)$ ，二分查找的时间复杂度为 $(\log_2 n)$ 所以总时间复杂度为 $O(n\log_2 n)$

### EX-2

思路：此题是一道要求输出逆序对个数的题目，根据OJ时间要求，应该使用一个时间复杂度小于 $O(n^2)$ 的算法。因此选择在归并排序的基础上修改算法来实现输出逆序对的个数。把输入数组分为左右两部分，分别从小到大排序，然后取出左边的第一个元素，即左数组的最大值，则此时逆序对计数为右数组中小于这个元素的元素个数。递归调用这个算法可以实现对于整个数组的逆序对个数的计数。由于可能逆序对个数很多，需要对每次递归的结果都模10000019

此算法是在归并排序的基础上修改而得来的，因此时间复杂度同归并排序，为 $O(n\log_2 n)$

### EX-3

思路：此题要求找到输入数组中两个大小相邻元素的差的最大值，因

此需要先排序。使用快速排序，选取中间元素，按照从小到大的顺序排序。然后再在排序好的数组中遍历一遍，直接找出相邻元素差的最大值。

快排的时间复杂度为 $O(n\log_2 n)$ ，遍历找差的最大值的时间复杂度为 $O(n)$ ，因此总的时间复杂度为 $O(n\log_2 n)$

## EX-4

思路：此题需要输出输入数组中定长片段中的最大值，由于时间限制，采用建立最大堆的方法，堆顶元素就是堆中元素的最大值，首先先建立大小为 $k$ 的堆，然后每多输入一个元素，直接加到堆底，堆的大小+1。然后利用上滤的方法直接比较这个结点和它的父结点的大小，依次上滤再次得到最大堆。由于只需要关注堆顶元素，因此只看堆顶元素什么时候会过期，若过期则将堆顶元素置0，并重新调用维护最大堆的函数来维护堆，使堆顶元素为没有过期的最大元素，0都在堆底。

建立最大堆的算法的时间复杂度为 $O(n\log_2 n)$ ，向堆中插入新元素并上滤的时间复杂度为 $O(\log_2 n)$ ，将过期的堆顶置0的时间复杂度为 $O(\log_2 n)$ ，由于插入元素和过期堆顶置0都是在for循环中，因此总的时间复杂度为 $O(n\log_2 n)$