problem2:

**算法1：**

采用堆排序的方法，需要两个步骤：

1. 先将n个元素形成一个优先级队列；
2. 通过对该优先级队列调用k次deQueue操作，能得到从最大到第k大的数据。

**复杂度分析：**

步骤1)需要O（N）的时间；

步骤2)中每次deQueue操作能选出该时刻队列中最大的元素，且时间均不超过logN时间，所以进行k次deQueue操作，能选出最大、第二大，…, 第k大的元素，时间消耗一共为klogN，最后一次选出的即为所要求找到的数据;

故算法1总时间消耗为O(N+klogN)

**算法2：**

建立一个容量为k的最小堆，始终保持k个最大的数据，当n个数据全部操作完毕后，堆顶元素即为第k大元素：

1）将n个数据的前k个，用enQueue的操作逐一加入，每次enQueue的时间复杂度一定小于logK，这样最终形成了拥有k个数据的最小堆；

2）将第k+1到第n个数据逐一与最小堆的堆顶比较，最终的堆顶为第k大数据：

如果小于堆顶，则忽略该数据；

如果大于堆顶，则用该数据替换原来堆顶，对新的堆顶进行percolateDownMin操作，向下过滤，调整最小堆，将新的最小值调整到堆顶；

**复杂度分析：**

步骤1）每次enQueue操作，最小堆元素个数始终小于等于k，步骤1的总复杂度小于klogk；

步骤2)从第k+1到第n个数据，每次操作，时间复杂度为1或logk，步骤2总复杂度一定小于(n-k)logk;

所以算法2的总复杂度为Nlogk；