

Heap / Priority_Queue

1. 题目要求：找第 K 大的数字，
2. 基本思路：
 - a) 维护一个容量为 K 的 heap。（因为 heap 就是一个非常厉害的数组，它的第一个数字就是数组的最小值， $O(\log(K))$ 内插入一个值， $O(\log(K))$ 内删除最小值）
 - b) 撸一遍输入之后，heap 的第一个就是答案。
3. 温馨提示：
 - a) Heap 有 minHeap 和 maxHeap 两种。分别求第 K 小，和第 K 大。
 - b) 实现 heap 的数据结构一般叫做：Priority Queue
4. 一个简约而不简单的例子：

215. Kth Largest Element in an Array

Find the **kth** largest element in an unsorted array. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Example 1:

Input: [3,2,1,5,6,4] and k = 2

Output: 5

Example 2:

Input: [3,2,3,1,2,4,5,5,6] and k = 4

Output: 4

Note:

You may assume k is always valid, $1 \leq k \leq \text{array's length}$.

答案在第二页。

解：

```
public class Solution {  
    public int findKthLargest(int[] nums, int k) {  
        PriorityQueue<Integer> largeK = new PriorityQueue<Integer>(k + 1);  
  
        for(int el : nums) { // 撸一遍  
            largeK.add(el);  
            if (largeK.size() > k) { // 始终维护容量为 k  
                largeK.poll();  
            }  
        }  
  
        return largeK.poll(); // 第一个就是我们想要的吧。  
    }  
}
```

5. 有奖抢答时间

- a) 用来 heap 后，算法时间和空间复杂度分别是啥？
- b) Heap 还有啥厉害的用法呢？

6. 习题：

老师课件上的：215，703，

考试里的：347，373，378，787，502，668，719，786，

7. 课外读物

<https://leetcode.com/problems/k-th-smallest-prime-fraction/discuss/115819/Summary-of-solutions-for-problems-%22reducible%22-to-LeetCode-378>

8. 这个话题有点沉重，加油。