

Description

Solution

Discuss (855)

Submissions

### 703. Kth Largest Element in a Stream

Easy

2308

1344

Add to List

Share

Design a class to find the  $k^{\text{th}}$  largest element in a stream. Note that it is the  $k^{\text{th}}$  largest element in the sorted order, not the  $k^{\text{th}}$  distinct element.

Implement `KthLargest` class:

- `KthLargest(int k, int[] nums)` Initializes the object with the integer `k` and the stream of integers `nums`.
- `int add(int val)` Appends the integer `val` to the stream and returns the element representing the  $k^{\text{th}}$  largest element in the stream.

#### Example 1:

##### Input

```
["KthLargest", "add", "add", "add", "add", "add"]
```

```
[[3, [4, 5, 8, 2]], [3], [5], [10], [9], [4]]
```

##### Output

```
[null, 4, 5, 5, 8, 8]
```

##### Explanation

```
KthLargest kthLargest = new KthLargest(3, [4, 5, 8, 2]);
kthLargest.add(3);    // return 4
kthLargest.add(5);    // return 5
kthLargest.add(10);   // return 5
kthLargest.add(9);    // return 8
kthLargest.add(4);    // return 8
```

#### Constraints:

- $1 \leq k \leq 10^4$
- $0 \leq \text{nums.length} \leq 10^4$
- $-10^4 \leq \text{nums}[i] \leq 10^4$
- $-10^4 \leq \text{val} \leq 10^4$
- At most  $10^4$  calls will be made to `add`.
- It is guaranteed that there will be at least `k` elements in the array when you search for the  $k^{\text{th}}$  element.

Java

Autocomplete

```
1  class KthLargest {
2
3      private int k;
4      private PriorityQueue<Integer> heap;
5
6      public KthLargest( int k, int[] nums ) {
7          this.k = k;
8          heap = new PriorityQueue<>();
9
10         for (int num : nums) {
11             heap.offer(num);
12         }
13
14         while (heap.size() > k) {
15             heap.poll();
16         }
17     }
18
19     public int add( int val ) {
20
21         heap.offer(val);
22
23         if (heap.size() > k) {
24             heap.poll();
25         }
26
27         return heap.peek();
28     }
29 }
30
31
32 /**
33  * Your KthLargest object will be instantiated and called as such:
34  * KthLargest obj = new KthLargest(k, nums);
35  * int param_1 = obj.add(val);
36  */
```