```
class KthLargest {

    PriorityQueue<Integer> pq;
    int size;
    public KthLargest(int k, int[] nums) {
        size = k;
        pq = new PriorityQueue<>();
        for (int i : nums) {
            pq.offer(i);
            if (pq.size() > size) {
                pq.poll();
            }
        }
    }

    public int add(int val) {
        if (pq.isEmpty() || pq.size() < size || pq.peek() < val) {
            pq.offer(val);
        }
        if (pq.size() > size) {
            pq.poll();
        }
        return pq.peek();
    }
}
```

Design a class to find the **k**th largest element in a stream. Note that it is the kth largest element in the sorted order, not the kth distinct element.

Your `KthLargest` class will have a constructor which accepts an integer `k` and an integer array `nums`, which contains initial elements from the stream. For each call to the method `KthLargest.add`, return the element representing the kth largest element in the stream.

Example:

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