KthLargest.java

```
1   package com.example;
2
3   import java.util.Arrays;
4   import java.util.PriorityQueue;
5   import java.util.Random;
6
7   public class KthLargest {
8       public int findKthLargest1(int[] nums, int k) {
9           PriorityQueue<Integer> minHeap = new PriorityQueue<>();
10  3       for (int i = 0; i < k; i++) {
11              minHeap.offer(nums[i]);
12          }
13
14  2       for (int i = k; i < nums.length; i++) {
15  2           if (nums[i] > minHeap.peek()) {
16                  minHeap.poll();
17                  minHeap.offer(nums[i]);
18              }
19          }
20
21  1       return minHeap.peek();
22      }
23
24      public int findKthLargest2(int[] nums, int k) {
25  1       int left = 0, right = nums.length - 1;
26          Random rand = new Random();
27          while (true) {
28  3           int pivot_index = left + rand.nextInt(right - left + 1);
29              int new_pivot_index = partition(nums, left, right, pivot_index);
30  2           if (new_pivot_index == nums.length - k) {
31  1               return nums[new_pivot_index];
32  3           } else if (new_pivot_index > nums.length - k) {
33  1               right = new_pivot_index - 1;
34              } else {
35  1               left = new_pivot_index + 1;
36              }
37          }
38      }
39
40      private int partition(int[] nums, int left, int right, int pivot_index) {
41          int pivot = nums[pivot_index];
42  1       swap(nums, pivot_index, right);
43          int stored_index = left;
44  3       for (int i = left; i < right; i++) {
45  2           if (nums[i] < pivot) {
46  1               swap(nums, i, stored_index);
47  1               stored_index++;
48              }
49          }
50  1       swap(nums, right, stored_index);
51  1       return stored_index;
52      }
53
54      private void swap(int[] nums, int i, int j) {
55          int temp = nums[i];
```

```
56          nums[i] = nums[j];
57          nums[j] = temp;
58      }
59
60      public int findKthLargest3(int[] nums, int k) {
61 1          Arrays.sort(nums);
62 2          return nums[nums.length - k];
63      }
64  }
```

Mutations

10	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
14	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
15	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
21	1. replaced int return with 0 for com/example/KthLargest::findKthLargest1 → KILLED
25	1. Replaced integer subtraction with addition → KILLED
28	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED 3. Replaced integer addition with subtraction → KILLED
30	1. Replaced integer subtraction with addition → KILLED 2. negated conditional → KILLED
31	1. replaced int return with 0 for com/example/KthLargest::findKthLargest2 → KILLED
32	1. changed conditional boundary → SURVIVED 2. Replaced integer subtraction with addition → KILLED 3. negated conditional → KILLED
33	1. Replaced integer subtraction with addition → KILLED
35	1. Replaced integer addition with subtraction → KILLED
42	1. removed call to com/example/KthLargest::swap → KILLED
44	1. changed conditional boundary → SURVIVED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
45	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED
46	1. removed call to com/example/KthLargest::swap → KILLED
47	1. Changed increment from 1 to -1 → KILLED
50	1. removed call to com/example/KthLargest::swap → KILLED
51	1. replaced int return with 0 for com/example/KthLargest::partition → TIMED_OUT
61	1. removed call to java/util/Arrays::sort → KILLED
62	1. Replaced integer subtraction with addition → KILLED 2. replaced int return with 0 for com/example/KthLargest::findKthLargest3 → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- com.example.KthLargestTest.testSort(com.example.KthLargestTest) (1 ms)

Report generated by PIT 1.5.0