Mastering Data Structures and Algorithms
Homework 4

Yiying Peng (Christine)

March 27, 2019

Problem 1

Given an unsorted array with N elements find the k-th highest given that k « N.

Sol. My Python algorithm implementation is as follows.

```
import queue


def kth_highest(arr, k):
    assert (len(arr) >= k)

    q = queue.PriorityQueue()
    for e in arr:
        if q.qsize() == k:
            top_e = q.get()  # smallest element in PriorityQueue
            max_e = max(top_e, e)
            q.put(max_e)
        else:
            q.put(e)

    print(q.get())  # k-th element, as the smallest (toppest) one in PriorityQueue


def main():
    solve([9, 1, 3, 4], 2)


if __name__ == '__main__':
    main()
```

I am using the idea of heapsort but with maximum k elements inside heap at one time. More details are as follows.

- When the heap is with size smaller than k, keep adding elements into heap.
- If the heap size equals k, compare the top one (smallest one) and the current one.
  - If the current one is larger than the top one, remove the top one and add the current one into the heap.
  - Else, continue.

Finally, the first element in that heap will be the kth-highest value in the original list.

The time complexity of this algorithm is $O(n \log k)$ instead of $O(n \log n)$ using sorting directly. The extra space complexity is $O(k)$.

□