EE360C: Algorithms
The University of Texas at Austin — Quiz #4
Dr. Christine Julien, Dr. Vallath Nandakumar — February 15, 2018 (*in class*)

Name: **EID:**

Quiz #4

Problem 1: Algorithm Design

Devise an algorithm for finding the k_{th} biggest element of an unsorted array of integers. Size of the array is n. Give your algorithm's time complexity and briefly explain why. Your algorithm's time complexity must be better than O(n log n).*(Hint: Use heap)*

Solution

Approach 1:
Firstly build a max heap, then pop k-1 elements out of it. After that, pop the k_{th} biggest element. The time complexity is O(n + k*log(n)), because time complexity is O(n) for building a heap and k*log(n) for popping out k elements.

Approach 2:
Using a min heap of size k+1 and adding elements of the array into the heap one by one. If the current heap contains k+1 elements, then pop the top element out of it before we add the next element. In this way, the heap would maintain k biggest elements seen so far. Keep doing this until all elements of the array have once been added to the heap and n-k elements have been popped out of the heap. Then the top element in the heap is the k_{th} biggest element. The time complexity is O(nlog(k)) because adding/removing a element into/out the heap would take log(k) time and there are at most 2n times operations of this.