CSC 225 FALL 2015
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 3
UNIVERSITY OF VICTORIA

1. Design a $O(n \log n)$-time algorithm to compute the number of inversions in a given array.

2. Suppose that your Quick-Sort algorithm uses the following pivot rule that picks the element in the "middle" - For an array $A[0, 1, \ldots, n-1]$ of size n, it uses the element in $A[n/2]$ as pivot if n is even and the element in $A[(n-1)/2]$ as pivot if n is odd. Give an input array of size 7, with values 1 to 7, on which your quick-sort algorithm that run the slowest.

3. Prove that it is impossible to develop a comparison-based implementation of the Priority Queue ADT such that both Insert and RemoveMin run in $O(\log \log n)$ time.

4. Suppose that we are given a sequence S of n elements, each of which is an integer in the range $[0, n^2 - 1]$. Describe a simple method for sorting S in $O(n)$ time. (Hint: Think of alternate ways of viewing the elements so that you can use Radix-Sort.)