Karl Bringmann and Vasileios Nakos **Summer 2020**

Sublinear Algorithms, Exercise Sheet 0

www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer20/sublinear-algorithms/

Total Points: **0** Due: Monday, **May 11**, 2020

This is a presence exercise sheet intended to be solved and discussed in the tutorial on Monday, May 11, 2020. It is not necessary to submit solutions to these exercises and there will be no points awarded. Nevertheless, these exercises are of similar flavor and difficulty than the upcoming compulsory exercises.

Exercise 1 — **0** points

In case you have not done so already, please read the "Primer on Randomness" (the document is linked on the course webpage).

Exercise 2 — **0** points

For starters, consider this real-world problem applicable to the current situation: In a group of n people, there are k persons infected with some kind of disease. There exist medical tests to examine individuals and check whether they are healthy or infected, but these tests are expensive.

Therefore, your task is to device a method to estimate the total number of infections k by using only a small number of tests. More precisely, with probability at least $1 - \delta$, you are expected to determine an approximation of k with absolute error at most εn using at most $O(\varepsilon^{-2} \log \delta^{-1})$ tests.

Exercise 3 — **0 + 0 + 0** points

Let $s_1, \dots, s_m \in [n]$ be a stream of m distinct elements. We say that an element s_i is of *rank* r if the number of stream elements smaller or equal than s_i is exactly r. In particular, the smallest element is of rank 1 and the largest element is of rank m. Observe that computing the median corresponds to computing an element of rank $\lfloor \frac{m}{2} \rfloor \leq r \leq \lceil \frac{m}{2} \rceil$.

1. In this exercise, we consider the relaxed problem of computing a $\frac{1}{3}$-approximate median: Given a stream of m elements, compute an element of rank $\frac{m}{6} \leq r \leq \frac{5m}{6}$. Design a randomized streaming algorithm for this problem with error probablity δ and space usage $O(\log \delta^{-1} \log(n + m))$.
2. So far you were allowed to assume that m is known in advance. Can you adapt your algorithm and not rely on that assumption anymore? That is, upon receiving an element your algorithm is expected to report a $\frac{1}{3}$-approximate median of all elements seen so far.
3. Consider the more general problem of computing an ε-approximate median: Given a stream of m elements, compute an element of rank $(\frac{1}{2} - \varepsilon)m \leq r \leq (\frac{1}{2} + \varepsilon)m$. Modify your algorithm to compute an ε-approximation. How does the space complexity change?

Exercise 4 — **0** points

In the previous exercise, you designed an efficient algorithm for computing an approximate median of a given stream. The goal of this exercise is to show that an approximation is essentially the best we can hope for. More specifically, prove that any *deterministic* streaming algorithm that *exactly* computes the median of a stream $s_1, \dots, s_m \in [n]$ requires $\Omega(m \log(n/m))$ bits of space.

Hint: Calculate and compare the following two quantities:

- *The number of possible states of the algorithm after receiving the first, say, $m/2$ elements.*
- *The number of "different" input streams of length $m/2$.*