Karl Bringmann and Vasileios Nakos **Summer 2020**

Sublinear Algorithms, Exercise Sheet 2

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

Total Points: **40** Due: 12:00 (noon), Monday, **June 15**, 2020

You are allowed to collaborate on the exercise sheets, but you have to write down a solution on your own, ***using your own words****. Please indicate the names of your collaborators for each exercise you solve. Further, cite all external sources that you use (books, websites, research papers, etc.).*

You need to collect at least 50% of all points on exercise sheets to be admitted to the exam.

Exercise 1 — 10 points

In this exercise we shall see a variant of the Heavy Hitters problem, to which we refer as the List-Heavy Hitters problem: Given a stream of m elements over the universe $[n]$, the task is to compute a set of size $O(\varepsilon^{-1})$ which is guaranteed to contain all elements occuring at least εm many times in the stream. The Misra-Gries algorithm is a deterministic algorithm for the List-Heavy Hitters problem:

Initialization: Maintain a set S of at most k index/counter pairs, initialized to $S = \emptyset$.

Update(i): There are three cases:

- If $(i, c) \in S$ for some counter c, then increment c: $S \leftarrow (S \setminus \{(i, c)\}) \cup \{(i, c + 1)\}$.
- Otherwise, if $|S| < k$, add $(i, 1)$ to the set: $S \leftarrow S \cup \{(i, 1)\}$.
- Otherwise, decrement all counters and remove zeros: $S \leftarrow \{(i, c - 1) : (i, c) \in S, c > 1\}$.

Query: Return $\{i : (i, c) \in S\}$ as the set containing all heavy hitters.

What's the (asymptotically) minimal choice of k for which the Misra-Gries algorithm works correctly? Give a correctness proof for that choice and analyze the space complexity of the algorithm.

Exercise 2 — 4 + 6 points

The goal of this exercise is to derive a streaming algorithm for counting triangles in a graph. More formally: Let G be an undirected graph with vertex set $[n]$ and edge set E. A *triangle* in G is a set of three distinct vertices $v_1, v_2, v_3 \in [n]$, which are pairwise connected: $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_1\} \in E$.

1. Let $x \in \mathbf{Z}^{\binom{n}{3}}$ be the vector indexed by size-3 subsets of vertices, where x_S equals the number of edges in S: $x_S = |\{e \in E : e \subseteq S\}|$. Recall that the p-th frequency moment of x is $F_p = \sum_S |x_S|^p$ for $p > 0$ and $F_0 = \lim_{p \to 0} F_p = |\{S : x_S \neq 0\}|$. Prove that

$$F_0 - \frac{3}{2}F_1 + \frac{1}{2}F_2$$

   equals the number of triangles in G.

2. In the usual streaming setup for graphs, the stream consists of the edges in an arbitrary order, but without repetitions. Design a streaming algorithm in that setting which counts the number of triangles in the streamed graph G with an absolute error of εn^3, e.g. using the first part of this exercise. Your algorithm is supposed to be correct with probability $\frac{2}{3}$ and the space usage should be bounded by $O(\varepsilon^{-2} \log n)$ bits.

For the following two exercises, recall the setup of combinatorial group testing from the lecture (or equivalently, the CountMin sketch framework):

Measurements: For each $r = 1, \dots, R$, we execute the following steps:

- Pick a random hash function $h_r : [n] \to [2k]$.
- Perform a measurement on each of the $2k$ groups $h_r^{-1}(b) = \{i \in [n] : h_r(i) = b\}$ for $b = 1, \dots, 2k$. We say that a group is infected if the measurement detected at least one infected individual in the group.

Find: Classify each $i \in [n]$ as infected if and only if it occured in infected groups only.

Exercise 3 — 10 points

In real world applications of combinatorial group testing, for various reasons it is often necessary to cap the group size at, say, $\leq d$. That restricted problem is called d-sparse combinatorial group testing. In this exercise, we examine how to deal with that restriction algorithmically: Prove that if $d \geq 50 \log n$, then $O(\max\{k, n/d\} \cdot \log n)$ measurements suffice to solve the non-uniform d-sparse combinatorial group testing problem – that is, any fixed set of k infected individuals can be recovered after $O(\max\{k, n/d\} \cdot \log n)$ measurements on groups of size $\leq d$, with probability $\frac{2}{3}$.

Hint: Use the same setup as in the lecture with $R = \Theta(\log n)$ rounds, but pick random hash functions $h_r : [n] \to [2\max\{k, n/d\}]$. For the analysis, you might find the multiplicative Chernoff bound (as stated e.g. in Exercise 1 on the previous sheet) useful.

Exercise 4 — 5 + 5 points

In the lecture, we proved that setting $R = \Theta(k \log n)$ we obtain a disjunct matrix, i.e. a set of measurements which allows uniform recovery for the combinatorial group testing problem. The goal of this exercise is to prove that the algorithm remains sound if instead of truly random functions, we pick h_r to be pairwise independent.

1. Fix distinct $i, i' \in [n]$. We shall refer to the number $|\{r : h_r(i) = h_r(i')\}|$ as the *common participation* of i and i'. Prove that with probability $1 - \frac{2}{3n^2}$, the common participation of i and i' is bounded by $(R-1)/k$ from above, for $R = \Theta(k \log n)$.
2. Prove that with probability $\frac{2}{3}$, if h_r is a pairwise independent hash function for each r (and independent across r) the set of measurements forms a disjunct matrix with probability $\frac{2}{3}$, allowing any set of k infected individuals to be correctly recovered.