Sublinear Time and Space Algorithms 2020B – Final Assignment

Robert Krauthgamer

Due: July 29, 2020 at 11:00

General instructions: Please keep your answers short and easy to read. You can use class material (results, notation, or references) without repeating it, unless asked explicitly to do so.

Specifically for this assignment: (1) Work on the problems completely by yourself, do not discuss it with other people or search for the solution online or at other sources, this is not the intention. (2) If you use any sources other than the class material (e.g., books, online lecture notes, wikipedia, or prior knowledge), point it out, even if you happened to find the solution online — I will not deduct points, but I want to know.

Part I (30 points)

Answer all of the following 3 "short" questions. Give short answers, that provide a justification/proof sketch/counter-example in 2-4 sentences, even for true/false questions.

A. Consider the frequency-vector model, where the stream contains additive updates to a vector $x \in \mathbb{R}^n$ whose coordinates are integers bounded by $\mathrm{poly}(n)$.

   Explain how to $(1+\epsilon)$-approximate $\sum_{i<j}(x_i + x_j)^2$ by a streaming algorithm with storage requirement $(\epsilon^{-1}\log n)^{O(1)}$ bits.

   Remark: As done in class, do not count storage of the algorithm's random coins.

B. Given $z \in \mathbb{R}^n$, let us "split" it as $z = z^1 + \ldots + z^{n/k}$, where z^1 has the k heaviest coordinates of z (and zero elsewhere), z^2 has the k next heaviest coordinates, and so forth. (Heaviest means largest absolute value; we assume that k divides n.)

   Is it true that:

$$\forall z \in \mathbb{R}^n, i \geq 2, \qquad \frac{\|z^i\|_1}{\sqrt{k}} \leq \|z^i\|_2 \leq \frac{\|z^{i-1}\|_1}{\sqrt{k}}.$$

C. Let H be a 2-universal family of hash functions $h : [n] \to [2M]$. Suppose we construct from it a new family H' of hash functions $h' : [n] \to [M]$, by "merging" every pair of buckets $\{2i-1, 2i\}$ for all $i \in [M]$.

   Is it true that H' is 2-universal?

   Remark: Formally, a random $h' \in H'$ is generated by the following process: pick $h \in H$ and set $h' = g \circ h$ (i.e., h' maps $x \mapsto g(h(x))$), where $g : [2M] \to [M]$ is a fixed "merging" function that maps $x \mapsto \lfloor \frac{x}{2} \rfloor$.

Part II (70 points)

Answer 2 of the following 3 questions.

1. Consider the frequency-vector model, where the stream contains additive updates to a vector $x \in \mathbb{R}^n$ whose coordinates are integers bounded by $\mathrm{poly}(n)$.

   Design a streaming algorithm to detect whether x is 2-sparse, i.e., distinguish whether $\|x\|_0 \le 2$ or not.

   Hint: Generalize the 1-sparsity detection algorithm seen in class for ℓ_0-sampling.

   Remark: As done in class, do not count storage of the algorithm's random coins.

2. Consider the frequency-vector model, where the stream contains additive updates to a vector $x \in \mathbb{R}^n$ whose coordinates are integers bounded by $\mathrm{poly}(n)$.

   Design a streaming algorithm that, given a query $i \in [n]$ (at the end of the stream), reports a $(1+\epsilon)$-approximation to $\|x_{[n]\setminus\{i\}}\|_2$ (the ℓ_2-norm of x when coordinate i is zeroed).

   Hint: Estimate the ℓ_2-norm of a virtual stream formed by subsampling the coordinates of x.

   Remark: As done in class, do not count storage of the algorithm's random coins.

3. In the Steiner Forest problem on a graph $G = (V, E)$, given k vertex-pairs $(u_1, v_1), \dots, (u_k, v_k)$, the goal is to find a subset of the edges $E' \subset E$ of minimum size, such that in the subgraph $G' = (V, E')$, each u_i is connected to its respective v_i. Notice that an optimal G' can have between 1 and k connected components.

   Design a streaming algorithm for the following restricted setting: The graph G is known in advance and fixed to be a complete binary tree $T = (V, E)$ with n leaves, hence $|V| = O(n)$, and the input stream contains k vertex-pairs $(u_1, v_1), \dots, (u_k, v_k)$. The algorithm should $(1+\epsilon)$-approximate the optimal size $|E'|$ (no need to report E' itself), with storage requirement of $(\epsilon^{-1} \log n)^{O(1)}$ bits.

   Hint: Since T is a tree, there is a unique path connecting u_i with v_i.

   Remark: As done in class, do not count storage of the algorithm's random coins.

THE END. Good luck!