Sublinear Time and Space Algorithms 2024A – Problem Set 4

Robert Krauthgamer

Due: February 28, 2024

General instructions: Please keep your answers short and easy to read. You can use results, calculations or notation seen in class without repeating them, unless asked explicitly to redo them.

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 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.

2. Suppose the input is a stream of updates (insertion and deletions) of the form (i, a), i.e., each item $i \in [n]$ arrives with a *unique* identifier $a \in [n^3]$. Let X be the final set (insertions minus deletions); to make sure it is well-defined, assume that (i, a) can be deleted only if it was inserted earlier. Uniqueness means that every two items (i_1, a_1) and (i_2, a_2) in X must have $a_1 \neq a_2$, but possibly $i_1 = i_2$.

   Design a sampler for such streams, that reports $(i^*, a^*) \in X$ such that i^* is drawn uniformly from the *distinct* values in $\{i : (i, a) \in X\}$.

   Hint: By ignoring the identifiers, the ℓ_0-sampler seen in class will report i^* with the correct distribution. Modify this sampler to report also an identifier of i^* at some occurrence in X.

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

   A motivating example: The stream describes packets seen by a router, where i is the packet's destination which can be repeated, and a identifies a flow (a sequence of packets sent by the same machine/program) that starts/ends.

3. We saw in class an algorithm (due to Feige) that, given a *connected* graph G and $\varepsilon \in (0, 1)$, estimates the graph's average degree d within factor $2+\varepsilon$ in time $O((\frac{1}{\varepsilon})^{O(1)}\sqrt{n})$.

   Show that this algorithm can be extended to r-uniform hypergraphs (see definitions below). Explain your modifications and the running time you obtain.

   Definitions: A *hypergraph* $G = (V, E)$ is a generalization of graphs where every hyperedge $e \in E$ is a subset (of arbitrary size) of the vertex set V. It is called *r-uniform* if every hyperedge $e \in E$ has cardinality $|e| = r$. Similarly to ordinary graphs, the *degree* of a vertex is the number of hyperedges containing it.

   Guidelines: Focus on small r, say $r = 4$. Explain the differences and do not repeat proofs that are the same. There is no need to optimize dependence on ε.