Sublinear Time and Space Algorithms 2016B – Final Assignment

Robert Krauthgamer

Due: July 4, 2016 at 13:00 (corrected version)

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.

Specifically for this assignment: (1) Work on the problems completely by yourself, without discussing it with other people. (2) If you use any sources other than the class material (e.g., books, articles, online lecture notes, or prior knowledge), point it out even if you happened to find the precise solution online — I will not deduct points, but I want to know.

1. Consider a streaming algorithm that estimates the ℓ_2-norm of $x \in \mathbb{R}^n$ by maintaining a linear sketch Ax and eventually reporting $\|Ax\|_2$, where A is a random $m \times n$ matrix constructed as follows. Pick $h : [n] \to [m]$ uniformly at random and set $A_{h(i),i} \in \{\pm 1\}$ at random with equal probabilities. Set all other entries of A to 0.

   Show that for suitable $m = (1/\varepsilon)^{O(1)}$, with probability at least 3/4 this algorithm reports a $(1+\varepsilon)$-approximation to $\|x\|_2$. Then compare the performance of this algorithm to the AMS tug-of-war algorithm with same success probability, in terms of storage requirement (ignoring the storage of random bits) and in terms of update time (notice that A above is sparse).

   Hint: Compute/bound the expectation and variance of $\|Ax\|_2^2$.

2. Consider the following rank problem, for "target" $t \in [m]$ and $\varepsilon \in (0, 1)$ known in advance. The input is a sequence $a_1, \ldots, a_m$ of items from the domain $[n]$, where we assume for simplicity that the items are all distinct. Define $\text{rank}(a_p) = |\{i \in [m] : a_i \leq a_p\}|$, for example, if m is odd the median item has rank $(m+1)/2$. The goal is to report an item a_p with $\text{rank}(a_p) \in [t \pm \varepsilon m]$.

   (a) Design a sampling algorithm that reads only $1/\varepsilon^{O(1)}$ items from the input, and succeeds with probability at least 3/4. Analyze the algorithm's success probability and query complexity.

   (b) Describe how to implement your sampling algorithm in the data-stream model, when the stream is $a_1, \ldots, a_m$ and its length m is *not* known in advance (but $m \leq n$ because they are distinct). Analyze the storage requirement (it should be polynomial in $\varepsilon^{-1} \log n$).

3. An unweighted graph G is called k-connected if every cut $(S, \bar{S})$ contains at least k edges.

   Design an algorithm that determines whether a dynamic graph G on vertex set $V = [n]$ (i.e., a stream of edge insertions and deletions) is 2-connected, using storage $\tilde{O}(n)$.

Hint: First verify that G is connected by constructing a spanning tree T. Then classify all possible cuts $(S, \bar{S})$ into those that contain two or more edges of the tree T and the rest, and finally use independent samples to verify whatever is still needed.

4. Prove that every randomized streaming algorithm that determines whether an input graph on the vertex set $V = [n]$ is connected, requires storage of $\Omega(n)$ bits.

   For full credit, prove it for undirected graphs, and under edge insertions only (no deletions).

   Hint: use reduction from indexing or from set disjointness.