Homework 2
Due Wed Oct 26, 5:00pm

Prof. Amit Chakrabarti
Computer Science Department
Dartmouth College

General Instructions: Please write concisely, but rigorously, and show your calculations explicitly, as we do in class. Each problem is worth 5 points, and only "nearly flawless" solutions will earn full credit.

Honor Principle: You are allowed to discuss the problems and exchange solution ideas with your classmates. But when you write up any solutions for submission, you must work alone. You may refer to any textbook you like, including online ones. However, you may not refer to published or online solutions to the specific problems on the homework. *If in doubt, ask the professor for clarification!*

Start Early! This homework is harder than the previous one. Start early, or else you won't be able to do it justice.

---

5. The first AMS algorithm for estimating the frequency moments F_k (i.e., not the "amazing F_2 algorithm") was described in class for vanilla streams only. Explain how it can be generalized to the cash register model; i.e., each token is of the form (j, c) — to be interpreted as the instruction "$f_j \leftarrow f_j + c$" — with c being a positive integer. Give careful pseudocode. Prove the correctness of your algorithm. In your proof, you may rely on the correctness proof we gave in class for the vanilla case: so don't reproduce those calculations, just explain what is new.

6. We have studied two algorithms for estimating the number of distinct elements in a stream, but neither of them produces a "sketch," as we have defined the term. Fix this.

   In greater detail: Consider a stream σ, in the turnstile model, that implicitly defines a frequency vector $\mathbf{f} = (f_1, \ldots, f_n)$. Suppose that, at all times during the processing of the stream and for all $j \in [n]$, we have $|f_j| \le m$. Design an algorithm that computes a sketch of σ so that, based on the sketch, one can quickly estimate the quantity $L_0(\sigma) := |\{j : f_j \ne 0\}|$. Your algorithm should use space $\mathrm{poly}(\log m, \log n, 1/\varepsilon)$, work *without* the assumption that $\mathbf{f} \ge 0$, and return an estimate that is off by no more than $(1 \pm \varepsilon)$, with probability at least 99% (say).

   Explain how your sketch can be used to estimate the number of places $j \in [n]$ where histograms (i.e., frequency vectors) corresponding to two streams σ and σ' differ.

   Hint: What happens to the ℓ_p-norm $\|\mathbf{x}\|_p$ of a vector $\mathbf{x} \in \mathbb{R}^n$ as $p \to 0$?

7. Consider a stream σ in the turnstile model, defining a frequency vector $\mathbf{f}$. The Count-Min Sketch solves the problem of estimating f_j, given j, but does not directly give us a *quick* way to identify, e.g., the set of elements with frequency greater than some threshold. Fix this.

   In greater detail: Let α be a constant with $0 < \alpha < 1$. We would like to maintain a suitable summary of the stream (some enhanced version of the Count-Min Sketch, perhaps?) so that we can, on demand, quickly produce a set $S \subseteq [n]$ satisfying the following properties w.h.p.: (1) S contains every j such that $f_j \ge \alpha F_1$; (2) S does not contain any j such that $f_j < (\alpha - \varepsilon)F_1$. Here, $F_1 = F_1(\sigma) = \|\mathbf{f}\|_1$. Design a data stream algorithm that achieves this. Your space usage, as well as the time taken to process each token and to produce the set S, should be polynomial in the usual parameters, $\log m$, $\log n$, and $1/\varepsilon$, and may depend arbitrarily on α.

   Hint: Suppose you combined all tokens in $\{1, 2, \ldots, n/2\}$ into one "supertoken", and similarly for all the other tokens in $\{n/2+1, n/2+2, \ldots, n\}$. Now, if you estimated the frequency of one of these supertokens to be less than αF_1, then you have eliminated $n/2$ candidates from S in one shot.

8. In class we motivated the second frequency moment F_2 as the size of a self-join: the join of a relation in a database *with itself*. In fact, one can design a sketch that can scan a relation in one pass (i.e., in streaming fashion) such that, based on the sketches of two *different* relations, we can estimate the size of their join. Explain how.

   Recall that for two relations (i.e., tables in a database) $r(A, B)$ and $s(A, C)$, with a common attribute (i.e., column) A, we define the join $r \bowtie s$ to be a relation consisting of all tuples (a, b, c) such that $(a, b) \in r$ and $(a, c) \in s$. Therefore, if $f_{r,j}$ and $f_{s,j}$ denote the frequencies of j in the first columns (i.e., "A"-columns) of r and s, respectively, and j can take values in $[n]$, then the size of the join is $\sum_{j=1}^{n} f_{r,j} f_{s,j}$.

   Hint: You have already seen the sketch in class! But you have to prove that it solves the above problem.