Homework Exercises on Streaming Algorithms

The maximum number of points for all exercises is 30. The grade for this homework set is: (number of scored points)/3.

Exercise Set Streaming I

Str.I-1 (4 points) Let $\sigma := \langle a_1, \ldots, a_m \rangle$ be a stream of m distinct items from the universe $[n]$. The *rank* of an item a_i is defined as follows:

$$\mathit{rank}(a_i) = 1 + \text{number of items in } \sigma \text{ smaller than } a_i.$$

Thus the smallest item has rank 1 and the largest item has rank m. A *median* of σ is a number of rank $\lfloor (m+1)/2 \rfloor$ or $\lceil (m+1)/2 \rceil$. Consider the following problem in the vanilla streaming model.

> MEDIAN: Given a stream $\sigma = \langle a_1, \ldots, a_m \rangle$ of m distinct items over the universe $[n]$, with $m < n/2$, compute a median of σ.

Either prove that any deterministic streaming algorithm that solves MEDIAN exactly must use $\Omega(m \log(n/m))$ bits in the worst case, or give a deterministic streaming algorithm that solves MEDIAN exactly using a sub-linear number of bits. If you give an algorithm, you should also prove its correctness and analyze the number of bits of storage it uses.

Str.I-2 (3 points) Consider Algorithm 8.1 from the Course Notes, which computes a set I containing all ε-frequent items in a stream. Suppose we change the algorithm as follows. If $|I| \geqslant 1/\varepsilon$ then, instead of decrementing the counters $c(j)$ of all $j \in I$, we only decrement the counters of all $j \in I$ with $c(j) = 1$. These counters are thus set to zero and the corresponding items j are removed from I. Someone claims that this algorithm still computes a set I containing all ε-frequent items; after all, each counter $c(j)$ that is not decremented stays closer to the true count of the number of occurrences of item j seen so far.

Prove or disprove this claim. To prove the claim, give a formal proof of the statement. To disprove the claim, give a concrete example of an input stream where the algorithm fails to report an ε-frequent item.

Str.I-3 (3 points) Consider the following sliding-window version of FREQUENT ITEMS. We are given an infinite stream $\sigma = \langle a_1, a_2, a_3 \ldots \rangle$ over the universe $[n]$ and a window size W, and we want to maintain a set I that contains all ε-frequent items (and possibly other items as well) within the current window. More precisely, after processing an item a_i, the following should hold. Define the window $\sigma(i, W)$ as

$$\sigma(i, W) \ := \ \begin{cases} a_{i-W+1}, \ldots, a_i & \text{if } i \geqslant W \\ a_1, \ldots, a_i & \text{if } i < W. \end{cases}$$

Let m_i denote the size of the window $\sigma(i, W)$; thus $m_i = \min(i, W)$. We define an item j to be *ε-frequent in $\sigma(i, W)$* if the number of occurrences of j in $\sigma(i, W)$ is at least $\varepsilon \cdot m_i$. Our goal is now to maintain a small set I such that, immediately after processing the token a_i we have: if j is an ε-frequent in $\sigma(i, W)$ then $j \in I$.

Describe a streaming algorithm for this problem that uses $O((1/\varepsilon) \log(n + W))$ bits.

Exercise Set Streaming II

Str.II-1 (2 + 2 points) Let $\sigma = \langle a_1, \ldots, a_m \rangle$ be a stream of m distinct items in the vanilla model. We wish to compute an element of rank $m/4$ in σ. Since this is hard to do exactly, we are satisfied with an item a_i such that $m/8 \leqslant \mathit{rank}(a_i) \leqslant 3m/8$.

(i) Give a streaming algorithm for this problem with the following properties: the probability that the rank of the returned item lies in the correct range is 218/512, the probability that the rank of the returned item is too small is 169/512, and the probability that the rank of the returned item is too large is 125/512. Prove that your algorithm has the desired properties and analyze its storage requirements.
NB: You may ignore rounding issues, and assume that an element chosen uniformly at random from the stream has probability 1/4 to lie in the correct range, and probability 1/8 and 5/8 that it is too small resp. too large.

(ii) Describe how to boost the success probability of your algorithm: present an algorithm that, for a given value $\delta > 0$, returns an item whose rank lies in the correct range with probability at least $1 - \delta$, and analyze the storage requirements of your algorithm.

Str.II-2 (2 points) Suppose that we want to compute a (1/10)-approximate median in a stream σ of m distinct items, and that we have a streaming algorithm that uses $O(\log(n+m))$ bits of storage and returns a (1/10)-approximate median with probability at least 0.05. Moreover, we know that the rank of the returned token never exceeds $\lceil m/2 \rceil$.

Explain how to boost the success probability of the algorithm so that it returns a (1/10)-approximate median with probability at least 0.95, and analyze the number of bits of storage used by your algorithm.

Str.II-3 (4 points) Give a 3-pass streaming algorithm that solves MEDIAN exactly and that, with probability at least 0.95, uses $O(\sqrt{m} \log n)$ bits of storage. Prove that your algorithm has the required properties. *Hint:* Use a random sampling approach. In the analysis the following inequality may be useful: $(1 - 1/k)^k < 1/2$ for all $k \geqslant 1$.

Exercise Set Streaming III

Str.III-1 (4 points) Prove that for $m < n$ any deterministic streaming algorithm that solves DISTINCT ITEMS exactly must use $\Omega(m \log(n/m))$ bits in the worst case.

Str.III-2 (2 points) Suppose person A has run the Count-Min sketch algorithm on a stream σ_1 with m_1 items, and person B has run the Count-Min sketch algorithm on a stream σ_2 with m_2 items. The items in both streams come from the universe universe $[n]$.

Now suppose we want to compute the Count-Min sketch for $\sigma_1 \circ \sigma_2$ from the sketch for σ_1 and the sketch for σ_2. Explain under which conditions this is possible, and explain how to compute the sketch for $\sigma_1 \circ \sigma_2$ in case the conditions are met. (Keep your answer short.)

Str.III-3 (4 points) Consider the CountMin sketch to estimate the frequencies of the items in a stream in the vanilla model. Suppose $\varepsilon = 0.2$ and $\delta = 0.5$ so that in the initialization phase we set $k = 10$ and $t = 1$. Give an example of an input stream σ such that the probability is very high that for at least one of the items $j \in \sigma$ the estimate of its frequency is much larger than its actual frequency. More precisely, give an example such that (for m large enough) the probability that there is an item j with $\widetilde{F}_\sigma[j] - F_\sigma[j] > m/2$ is at least 0.99.