CS 217 – Algorithms Design and Analysis
Homework Assignment 6

Shanghai Jiaotong University, Fall 2015

Handed out on Wednesday 2015-12-23

Due on Thursday 2015-12-31

You can hand in your solution either as a printed file or hand-written on paper (in this case please *write nicely*). You have to justify your solutions (i.e., provide proofs).

You can solve the homework assignment in your group, and every group should hand in *one* solution. Do not copy solutions from other groups! If you are completely stuck, you may ask me for advice!

6 Deterministic Streaming Algorithms

Suppose you are processing a data stream $a_1, \ldots, a_m$ of elements from a universe $[n]$. You want to answer the following question:

> Is there a majority element? That is, is there some $b \in [n]$ appearing more than $m/2$ times?

Theorem 1. *Suppose A is a deterministic streaming algorithm that correctly answers the above question, i.e., decides whether there is a majority element. Suppose $n \geq m$. Then A needs at least $m - \log(m)$ bits of memory.*

Proof. Our algorithm has s bits of memory. Thus, the memory can be in $S := 2^s$ different states. Let $(a_1, \ldots, a_k) \in [n]^*$. Since our algorithm is deterministic, it will be in a certain state $f(\mathbf{a}) \in [S]$ after processing the elements $a_1, \ldots, a_k$ (in that order).

Now let $A, B \in [n]$ be two distinct sets of $\lfloor m/2 \rfloor$ elements. So there is some $a \in A \setminus B$ and some $b \in B \setminus A$. Let $\mathbf{a}$ be a data stream consisting of the $|A|$ elements of the set A, in ascending order, say. Similarly define $\mathbf{b}$.

Claim: $f(\mathbf{a}) \neq f(\mathbf{b})$. If not, then $f(\mathbf{a}) = f(\mathbf{b})$, i.e., these two strings leave the algorithm in the same state, and therefore also $f(\mathbf{ac}) = f(\mathbf{bc})$ for every sequence $\mathbf{c} \in [n]^*$. Now set $\mathbf{c} := aa \ldots a$, of length $\lceil m/2 \rceil$. Then $\mathbf{ac}$ has a majority element (namely a), but $\mathbf{bc}$ has not. However, since $f(\mathbf{ac}) = f(\mathbf{bc})$ the algorithm gives the same answer on both strings and is therefore incorrect. This proves the claim.

Now that the claim is proved, observe that f defines an injective function from $\binom{n}{\lfloor m/2 \rfloor}$ into $[S]$. Therefore $S \geq \binom{n}{\lfloor m/2 \rfloor}$. Since $n \geq m$ this is greater than $\binom{m}{\lfloor m/2 \rfloor} \geq 2^m/m$, and therefore $s = \log_2(S) \geq m - \log_2(m)$. Our algorithm needs at least $m - \log_2(m)$ bits of memory. $\square$

Exercise 6.1. We consider another streaming problem:

> Given a data stream $a_1, \ldots, a_m$, compute the number of distinct elements. That is, compute $|\{a_1, \ldots, a_m\}|$.

Show that any deterministic algorithm solving the above problem requires at least n bits of memory.

6.1 Second and Fourth Moment

Let f_i be the number of times element i appears. That is,

$$f_i := |\{1 \leq j \leq m | a_j = i\}| \ .$$

In the lecture we saw the algorithm by Alon, Matias, and Szegedy to estimate the "second moment" of the data stream:

$$F_2 := \sum_{i=1}^{n} f_i^2 \ ,$$

where $f_i := |\{1 \leq j \leq m | a_j = i\}|$ is the frequency of element i. Here is a naive way to generalize their idea in an attempt to estimate the fourth moment:

$$F_4 := \sum_{i=1}^{n} f_i^4 \ .$$

Let $\sigma : [n] \to \{-1, 1, -i, i\}$ be 8-wise independent (yes, we are using complex numbers here).

Exercise 6.2. Let $X := \sum_{k=1}^{n} f_k \sigma(k)$.

1. Show how to compute X with $\log(m)$ bits (there will be $8 \log(n)$ bits to store σ, but you can simply take σ as granted).
2. Show that $\mathbb{E}[X^4] = F_4$, so X^4 is really an unbiased estimator of F_4.
3. How would you check whether X^4 is a good estimator? What goes wrong with our naive approach?

Exercise 6.3. Consider the following problem:

> Given a data stream $a_1, \ldots, a_m$ of elements from $[n]$, output some element that is not in $\{a_1, \ldots, a_m\}$.

1. Suppose $m \leq n/1000$. Give a randomized algorithm that either outputs "?" or outputs an element $a \in [n] \setminus \{a_1, \ldots, a_m\}$. For any input sequence $a_1, \ldots, a_m$ the probability of "?" must be at most $1/100$ and your algorithm must use at most $O(\log n + \log m)$ space. **Hint.** This has a really really simple solution!
2. Suppose $m \leq \sqrt{n}$. Can you give a deterministic algorithm for this problem? **Remark.** I have absolutely no idea. I haven't even searched the literature.