15-451/651 Algorithms, Fall 2014

Homework # 8 **Due: Thursday November 13, 2014**

Follow the rules of the previous written assignment. The first to solve problem H gets $20.

(50 pts) 1. **Streaming Medians**

In this problem we develop the algorithm to find an approximate median using a sampling idea.

Given a set A of n distinct numbers, let M be the set of elements with ranks in the interval $(\,(1-\varepsilon)\frac{n}{2}, (1+\varepsilon)\frac{n}{2}\,)$, let L be the elements with ranks $[1, (1-\varepsilon)\frac{n}{2}]$, and H be the elements with ranks $[(1+\varepsilon)\frac{n}{2}, n]$. An ε-*approximate median* is any element in M.

(a) Let $\varepsilon > 0$. Suppose you have a coin with "heads" probability $p = \frac{1}{2}(1-\varepsilon)$ and "tails" probability $1-p = \frac{1}{2}(1+\varepsilon)$. You flip it $\ell = 2k+1$ times. Show that the probability of getting a majority of flips being heads (i.e., at least $k+1$ heads) is at most

$$\frac{1}{\varepsilon^2}(1-\varepsilon^2)^{k+1} \leq \frac{1}{\varepsilon^2}\, e^{-\varepsilon^2(k+1)} \leq \frac{1}{\varepsilon^2}\, e^{-\varepsilon^2 k}.$$

(Hint: write down the exact probability and then use simple approximations. This is not the best answer possible, we know how to do better; if you can do better, please do not panic.)

(b) Consider the algorithm:

Define $k := \frac{1}{\varepsilon^2} \ln \frac{2}{\varepsilon^2 \delta}$.

Let S be a set of $\ell = 2k+1$ uniformly random elements of A (chosen with replacement). Let m be a median of S. Return m. (Observe that we succeed exactly when m lies in M.)

Show that $\mathbf{Pr}[m \in L \cup H] \leq \delta$. (Hint: Show that $\mathbf{Pr}[m \in L] \leq \delta/2$.)

Hence, observe that the algorithm above finds an ε-approximate median with probability at least $1-\delta$. (This is not a streaming algorithm, however.)

(c) (Nothing to do here.) Suppose T is a uniformly random subset of A, of size ℓ. (Hence T is like sampling ℓ random elements *without replacement*.) Return the median m' of T. We're not asking you to prove it, but it is possible to show that $\mathbf{Pr}[m' \in L \cup H] \leq \delta$, even in this setting.

(d) Give a procedure that, given a stream $a_1, a_2, \ldots,$ of numbers, maintains at each time $t \geq \ell$ a set $T \subseteq a_{[1:t]}$ with size exactly ℓ, such that T is a uniformly random subset of $a_{[1:t]}$ of size ℓ. (This procedure stores at most ℓ numbers and time t in memory.)

Putting the parts together, observe that if we run the procedure in part (e) to maintain the random set T of size $O(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon^2 \delta})$, the median of the elements in T at some time t is an ε-approximate median of $a_{[1:t]}$ with probability at least $1-\delta$.

(20 pts) 2. **Counting Substrings**

A suffix tree has been built for a string s of length n. (Actually the suffix tree has been built for the string $s\$$ which is s augmented with a special unique terminal character.) Your job is to give an algorithm which counts the number of distinct non-empty substrings of s in $O(n)$ time.

For example, if $s =$ `abab`, then there are seven such substrings: `a`, `ab`,`aba`,`abab`,`b`,`ba`,`bab`.

(30 pts) 3. **LDIS**

The Longest Duplicate Initial Substring problem (LDIS) is the following. Given a string s compute the length of the longest string w such that ww occurs at the beginning of s (or determine if no such string exists). (Note that ww means the string w concatented with itself.)

For example if $s =$`aaaaabaaaab` then the answer is 2.

(a) Give a linear-time probabilistic algorithm for this problem based on Karp-Rabin fingerprinting.

(b) Give a linear-time algorithm for this problem based on suffix trees.

($20) H. **Problem To Be Announced**