MA421 Advanced Algorithms Lent 2019

Exercises 9

1. Implement the AMS estimation algorithms for F_2 and F_k, for general k, (F_k being the kth frequency moment) from the lectures in Java. Note that a reliable estimation in the latter algorithm was achieved by taking the median of averages of copies of a basic estimator. For the purposes of this exercise, you are only required to output the average of multiple copies of the basic estimator. Your program should receive the input parameters and the data stream from the command line as
`>java FreqMom k t a1 a2 ... am`,
where each argument is an integer, `k` is the order of the frequency moment of the stream to estimate, `t` is the number of the basic estimators to average, and `ai`'s are the elements of the data stream. The program should output the final estimation and also the actual value (that is, the k-th frequency moment) for the purpose of comparison. Feel free to implement the median-of-averages technique as well to check whether the quality of the estimation improves. You can also improve the F_2 algorithm (in terms of space complexity) by using 4-wise independent samples (as described in Question 3 below).

2. Let g be a function such that $g(0) = 0$. Let f_i, for $i = 1, \ldots, n$, be the frequency of value i in a given data stream. Suppose we would like to estimate

$$\sum_{i=1}^{n} g(f_i),$$

using a data stream algorithm. Describe an algorithm such that the output of the algorithm is unbiased: that is, the expected value of the output is $\sum_{i=1}^{n} g(f_i)$.

3. In the description of AMS algorithm for estimating F_2 from the lecture, the algorithm generates n independent random bits $r_1, r_2, \ldots, r_n$ to use throughout the execution. Obviously, such space usage is against the spirit of streaming algorithms. However, we can observe that, in the analysis, we only needed any four of these bits to be independent (as opposed to all of them being independent) for a good bound on the variance of the estimator. Hence, we can improve the space complexity if we can generate random numbers accordingly.

   Definition 0.1 *Let $r_1, r_2, \ldots, r_n$ be identically distributed random variables with each r_i chosen uniformly at random from set U. Then, $r_1, r_2, \ldots, r_n$ are* k-wise independent *if, for any $S \subseteq \{1, \ldots, n\}$ such that $|S| = k$ and any $\rho \in U^k$, we have*

$$\Pr[r_S = \rho] = \frac{1}{|U|^k},$$

   where r_S is the vector defined by r_i such that $i \in S$.

   Let U be $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ for some prime number $p \geq n$. Let a, b chosen independently and uniformly at random from U. Now, define, for $i \in U$,

$$r_i = a \cdot i + b,$$

where the multiplication and the addition is done modulo p (hence, over the field $\mathbb{F}_p$). Show that $r_1, r_2, \ldots, r_n$ are pairwise independent. What is space complexity needed to generate such r_i during the execution of a streaming algorithm. How would you extend this to 4-wise independent random variables?

4. Consider the following sliding-window version of FREQUENT-ITEMS problem on a data stream. (You can read about the standard version of the FREQUENT-ITEMS over the whole data stream in MA421 Summer 2014 Exam, Question 4, available with solutions on Moodle.) We are given an infinite data stream $\sigma = \langle a_1, a_2, a_3, \ldots \rangle$ over the universe $\{1, 2, \ldots, n\}$ 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

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

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 $\epsilon \cdot m_i$. Our goal is now to maintain a small set I such that, immediately after processing the token a_i, the following holds: if j is ϵ-frequent in $\sigma(i, W)$, then $j \in I$. Describe a streaming algorithm for this problem that uses $O((1/\epsilon)(\log n + \log W))$ bits. Prove that your algorithm is correct and satisfies the required space complexity.

Solutions to the second question is to be handed in at the lecture on **19 March 2019**. The Java code for Question 1 should be emailed to `t.batu@lse.ac.uk`.

