HW 4

Due on **Monday, April 1, 2024 at 10am**

CS 574: Randomized Algorithms, Spring 2024 Version: **1.0**

Submission guidelines same as previous homework.

10 (100 PTS.) Streaming inverse frequencies.

Let S be a stream of m numbers taken out of the set $\{1, \ldots, n\}$. For a number $i = 1, \ldots, n$, let f_i be the number of times that i appears in the stream. You can assume that this stream contains at least $m/10$ distinct values.

You are also given parameter $\varepsilon \in (0,1)$ and $\delta \in (0,1)$. Design an algorithm that uses small space (in both n and m) that outputs an estimate H to the quantity $G = \sum_{i:f_i \neq 0} \frac{1}{f_i}$. The algorithm is required to have the property that

$$\mathbb{P}[(1-\varepsilon)G \leq H \leq (1+\varepsilon)G] \geq 1-\delta.$$

Provide full details!

(Hint: Follow the algorithms seen in class. Figure out where you have to use the provided assumption.)

11 (100 PTS.) Estimate it.

Let $S = s_1, s_2, \ldots, s_m$ be a stream (m is not known to you in advance). Given an item s_i in the stream, you can check if it is valid by calling an oracle D. Let τ be the number of valid elements in the stream. The oracle calls are expensive. Given parameters ε, δ, describe an algorithm that outputs an estimate u for the number of items in the stream that are valid, such that $\mathbb{P}[\tau - \varepsilon m \leq u \leq \tau + \varepsilon m] \geq 1 - \delta$.

Your algorithm needs to use little space, and few oracle calls. In particular, how many oracle calls does your algorithm performs, say in expectation. How much space does your algorithm uses?

12 (100 PTS.) Streaming for kth smallest number.

You are given as input a parameter k.

12.A. (10 PTS.) Describe a streaming algorithm that output the kth smallest number in a stream S of n (distinct) numbers $s_1, \ldots, s_n$. How much space does your algorithm use.

12.B. (70 PTS.) Given a parameter $\varepsilon \in (0,1)$, describe a streaming algorithm that outputs a number t, such that $(1-\varepsilon)k \leq \mathrm{rank}(S,t) \leq (1+\varepsilon)k$, where $\mathrm{rank}(S,t)$ is the rank of the number of t in S (i.e., it is the number of elements in S that are $\leq t$).

For credit, your solution should use space independent of k, and succeed with probability $\geq 1 - 1/n^{10}$. Prove that your algorithm succeeds with the stated probability.

12.C. (20 PTS.) (Probably hard and tedious.) Solve the previous part, under the settings where you do not know n in advance (i.e., the length of the stream). Instead, you are given an additional parameter $\delta \in (0,1)$, and your streaming algorithm needs to succeed with probability $\geq 1-\delta$. How much space does your streaming algorithm use?