Assignment 2

COMP 3801- Fall 2024

1 Instructions

Each question is worth 10 marks. The assignment is due in Brighspace before the due date. Please write clearly and answer questions precisely. As a thumb rule, the answer should be limited to ≤ 2 written pages, with ample spacing between lines and in margins, for each Question. Always start a new question on a new page, starting with Question 1, followed by Question 2, ..., Question n. Please cite all the references you have used/consulted as the source of information for each question. If a question asks you to Prove or Show, please clearly spell out the proof - the technique used - and show each step of the proof. Don't expect (partial) marks if the main idea isn't clear to us.

2 Problems

1. In the algorithm for computing the number of 1s in a sliding window of size N, we had at most two buckets of each type i, where $i = 0, 1, \ldots, \lceil \log N \rceil$. Show what part of the analysis won't work if we restrict at most one bucket of each type.

2. Consider the following minor variation of the data structure for approximately counting number of 1s in a binary stream among the last N bits received. Let $r \geq 2$ be an integer parameter. In place of maintaining two buckets of type B_i let us maintain r or $r-1$ copies of B_i for each $i \geq 1$. Note that the buckets of type B_0 may be less than $r-1$. Updates are as before, and at any time we exceed r copies of any type of buckets, we take the oldest two buckets and merge them to form a new bucket of the next size. For the query assume that the bucket labeled B_j is only partially overlapping the query window. Show the following.

   (a) Show that at least $1 + \sum_{i=1}^{j-1} (r-1)2^i$ 1-bits are in the query window.

   (b) Argue that the true count and the reported value of the count are within a factor of $1 \pm \frac{1}{r-1}$.

   (c) Show that by setting $r = 1 + \frac{1}{\epsilon}$, we can obtain a data structure of size $O(\frac{1}{\epsilon} \log^2 N)$ that can approximate the count of the number of 1s within a factor of $1 \pm \epsilon$.

   (d) Determine the value of r, where we use up to $r \geq 2$ buckets of type B_i for $i \geq 0$, so that the count of the number of 1's reported by the algorithm is within 5% of the actual count of 1s among the last N bits seen in a data stream. Justify your choice of r.

3. Assume that we have a stream consisting of numbers from the set $\{-1, 0, +1\}$ and we are interested in maintaining the sum of last N bits of the stream. In this exercise we will show

that it will require $\Omega(N)$ bits to maintain an approximate sum that is within a constant factor of the exact sum. Suppose we have an algorithm $\mathcal{A}$ that maintains the approximate sum within a constant factor on the input consisting of stream of $\{-1, 0, +1\}$. Assume that we have a bit string X consisting of $\frac{N}{2}$-bits composed of only 0s and 1s. We form an input of N bits for Algorithm $\mathcal{A}$ as follows: We replace each 0-bit of X by a pair of bits $(1, -1)$ and each 1-bit of X by the pair $(-1, 1)$. Now this sequence of N-bits is presented to our algorithm $\mathcal{A}$. Note that the exact sum of these N-bits is 0. In addition to these N bits derived from stream X, the next set of N bits that will be presented to $\mathcal{A}$ consists of only 0-bits and we will show that we will recover completely the original bit vector X. Answer the following:

(a) Show that if the $(N + 1)$-st bit in the stream for $\mathcal{A}$ is 0, the output to the exact sum query on receiving this bit will be $+1$ (respectively -1) if and only if the 1st bit in the stream of X was a 1 (respectively, 0). Moreover, on receiving this $(N + 1)$st 0 bit, Algorithm $\mathcal{A}$ will output a positive number (resp. negative number) if and only if the 1st bit in the stream of X was a 1 (respectively, 0)

(b) For a positive integer $i < \frac{N}{2}$, show that after receiving the $(N + 2i - 1)$-th 0 bit, the output to the approximate sum query algorithm $\mathcal{A}$ is a positive number (respectively a negative number) if and only if the i-th bit in the stream X was a 1 (respectively, 0).

(c) Show that after receiving the $2N$-th 0 bit by $\mathcal{A}$, we would have completely recovered all the bits of the stream X (and therefore the first N bits of the stream A).

(d) Conclude that to estimate the approximate sum within a constant factor in a sliding window of size N in a stream of (positive and negative) numbers we need to store $\Theta(n)$ bits.

4. Consider tracking the most popular movies from the sale of movie tickets sold worldwide. Let $c = 10^{-6}$ and $\tau = 1/2$. We maintain decaying scores for movies whose threshold is at least τ. For each new ticket sale for a movie, say without loss of generality this is for the movie M, perform the following steps.

(a) For each movie whose score is being maintained, its new score is reduced by a factor of $(1 - c)$. (To be precise, if the score of a movie was s, the new score is $s := s(1 - c)$.)

(b) If we have the score of M, add 1 to that score. Otherwise, create a new score for M and initialize it to 1.

(c) Remove any score that falls below τ.

Answer the following questions:

(a) What is the sum of all scores at any point in time?

(b) How many scores are maintained at any given time?

(c) If $\tau = 1/3$ instead of $1/2$, what will be the number of scores maintained and the sum of all the scores at any point in time?

5. Let A be a data stream. Without loss of generality, assume that $a_1, a_2, \ldots, a_k$ are the most frequent k elements with frequencies $f_1 \geq f_2 \geq \cdots \geq f_k$, respectively. We sample each element of A uniformly at random to construct a multi-set A'. We are interested to know how many times we need to sample A so that the most frequent k elements have a representative in A'

with high probability. In other words, what should be the size of A' so that with probability $\geq 1-\epsilon$, for $\epsilon > 0$, so that $a_1, \ldots, a_k \in A'$.
Hint: Let us assume that $s = |A'|$. Estimate first the probability that if we choose s elements from A, each uniformly at random with replacement, what is the probability that $a_k \notin A'$? What is an upper bound on the probability that none of $a_1, a_2, \ldots, a_k$ are in A'? Show that by choosing $s = O(\frac{|A|}{f_k} \log \frac{k}{\epsilon})$, with probability $\geq 1-\epsilon$, $a_1, \ldots, a_k \in A'$.

6. Recall the Tug of War algorithm for estimating frequency moment F_2 for a data stream $A = (a_1, \ldots, a_m)$, where each $a_i \in A$ is drawn from the universe U:

Algorithm - Tug of War using hash function $h : U \to \{+1, -1\}$
Step 1: Initialize $Y := 0$.
Step 2: For each element $x \in U$, evaluate $r_x = h(x)$.
Step 3: For each element $a_i \in A$, $Y := Y + r_{a_i}$
Step 4: Return $\hat{F}_2 = Y^2$

Let $h_1, \ldots, h_k$ be k independent hash functions, where each $h_j : U \to \{-1, +1\}$. Suppose, we execute the algorithm k times, resulting in $Y_1^2, Y_2^2, \ldots, Y_k^2$, where the j-th run uses the hash function h_j. I.e.,

j-th Run of Tug of War Algorithm using $h_j : U \to \{-1, +1\}$.
Step 1: Initialize $Y_j := 0$.
Step 2: For each element $x \in U$, evaluate $r_x = h_j(x)$.
Step 3: For each element $a_i \in A$, $Y_j := Y_j + r_{a_i}$
Step 4: Return Y_j^2

Set $\bar{Y}^2 = \frac{1}{k} \sum_{j=1}^{k} Y_j^2$

Show the following:

(a) $E[\bar{Y}^2] = F_2$
(b) $Var[\bar{Y}^2] = \frac{1}{k} Var[Y^2]$
(c) $Pr\left(|\bar{Y}^2 - E[\bar{Y}^2]| \geq \sqrt{\frac{2}{k}} c E[\bar{Y}^2]\right) \leq \frac{1}{c^2}$
(d) Given a positive constant $0 < \epsilon \leq 1$, find an appropriate value of k so that $Pr\left(|\bar{Y}^2 - E[\bar{Y}^2]| \geq \epsilon c E[\bar{Y}^2]\right) \leq \frac{1}{c^2}$.

7. Let $A = (a_1, \ldots, a_m)$ be a data stream of m elements, where each $a_i \in N = \{1, \ldots, n\}$. Let k be a positive integer. The k-th frequency moment F_k is defined as $F_k = \sum_{i=1}^{n} m_i^k$, where m_i is the number of times i appears in A. In this exercise, we will provide an estimate for F_k. Choose an index $p \in \{1, \ldots, m\}$ uniformly at random. Define r to be the number of times the element a_p occurs in the stream among the elements $(a_p, a_{p+1}, \ldots, a_m)$. Define the random variable $X = m(r^k - (r-1)^k)$. Answer the following.

(a) Let $A = (1, 2, 2, 3, 1, 1, 2)$. Evaluate F_k and $E[X]$.

(b) Show that $E[X] = F_k$.

(c) Show that using $O(\log n + \log m)$ space, we can approximate F_k.

8. Recall the min-hash signature matrix M with $n = br$ signatures, where we made b bands, each consisting of r-rows. We identify that a pair (S, T) of documents are similar if the signatures of S and T match exactly in at least one of the bands. We say that the signature matches in a band if the signatures are identical in each of the rows of that band. This computation performs an AND operation within a band and an OR operation among the bands. We estimated that the probability that S and T will be declared similar is $f(s) = 1 - (1 - s^r)^b$, where s is the Jaccard similarity of S and T. Suppose we reverse the role of AND and OR, i.e., now we say that the two documents are similar if the signatures match in each of the bands, and within the band the signatures match if they match in at least one of the rows of that band. What is the probability of determining whether the two documents are similar in this case (in terms of the parameters b, r, and s).

9. Let us assume that we have a large collection B of binary vectors in dimension $d = 100,000$. We are asked to compute a data structure so that the following queries can be answered efficiently. Given any query binary vector q in dimension d, we are interested to report all the binary vectors in B that are approximately 96% similar to q. We say that two vectors $a = a_1 a_2 \ldots a_d$ and $b = b_1 b_2 \ldots b_d$ are 95% similar if $a_i = b_i$ for at least 95% of indices i, $1 \leq i \leq d$. Design an algorithm that computes such a data structure and show how each query can be answered efficiently. The time to answer the query q should not exceed $O((k+1)d)$, where k is the number of vectors in B that are at least 95% similar to q. It is fine if you have some false positives and negatives, but their percentage shouldn't be large.

10. Answer the following problems on matrices and provide justification.

    (a) Let A be $n \times n$ square matrix consisting of real numbers, and let I be an identity matrix of dimension $n \times n$. In your own words, explain why the eigenvalues of A can be obtained by determining the n possible values of λ that satisfy the equation Determinant$(A - \lambda I) = 0$. Please refer to any textbook in linear algebra to review identity matrices and determinants.

    (b) For a square matrix A show that

        i. The product of its eigenvalues equals the determinant of A. (Hint: Consider the characteristic polynomial and set $\lambda = 0$.)

        ii. The sum of its eigenvalues equals the sum of the diagonal entries of A (called the *trace* of A).

        iii. The eigenvalues of A are same as that of A^T. Do they have the same eigenvectors?