Homework II, Advanced Algorithms 2020

Due on Friday May 29 at 17:00 (upload one solution per group on moodle). Solutions to many homework problems, including problems on this set, are available on the Internet, either in exactly the same formulation or with some minor perturbation. It is *not acceptable* to copy such solutions. It is hard to make strict rules on what information from the Internet you may use and hence whenever in doubt contact Michael Kapralov.

1 (25 pts) Suppose that you are given an insertion only stream, i.e. sequence of m numbers $i_1, \ldots, i_m$, where each number is between 1 and n (we assume that $m = n^{O(1)}$). Let $x \in \mathbb{R}^n$ denote the resulting frequency vector, i.e. for every $j \in [n] := \{1, 2, \ldots, n\}$ we let x_j denote the number of occurrences of j in the stream.

Give a randomized algorithm for finding the approximate median of a stream, i.e. for finding an element i such that

$$\sum_{j=1}^{i} x_j \geq (1/2 - \epsilon)m$$

and

$$\sum_{j=1}^{i-1} x_j \leq (1/2 + \epsilon)m$$

for a parameter $\epsilon > 0$. For a failure probability $\delta \in (0,1)$, your algorithm should use $O(\frac{1}{\epsilon^2} \log n \cdot \log(1/\delta))$ bits of space. You may assume that the length of the stream m is given to you as a parameter, and that your algorithm has access to a source of random coins.

2 (30 pts) Suppose that you are given an insertion only stream, i.e. sequence of m numbers $i_1, \ldots, i_m$, where each number is between 1 and n (we assume that $m = n^{O(1)}$). Let $x \in \mathbb{R}^n$ denote the resulting frequency vector, i.e. for every $j \in [n] := \{1, 2, \ldots, n\}$ we let x_j denote the number of occurrences of j in the stream.

Recall that the AMS sketch approximates $||x||_2^2$ by maintaining Ax for a $m \times n$ matrix with random entries in $\{\pm\frac{1}{\sqrt{m}}\}$, and outputting $||Ax||_2^2$, which satisfies

$$(1-\epsilon)||x||_2^2 \leq ||Ax||_2^2 \leq (1+\epsilon)||x||_2^2$$

with probability at least 2/3 if $m = C_1/\epsilon^2$ for an absolute constant $C_1 > 0$. Note that the time required to maintain Ax is C_1/ϵ^2 per update in the stream, as the matrix A is dense. In this problem we design an alternative matrix Π such that $||\Pi x||_2^2$ approximates $||x||_2^2$ well, and is cheaper to maintain.

Choose a pairwise independent hash function $h : [n] \to [m]$, and a four-wise independent hash function $\sigma : [n] \to \{-1, +1\}$. Define an $m \times n$ matrix Π by letting, for each $j \in [n] = \{1, 2, \ldots, n\}$

$$\Pi_{ij} = \begin{cases} \sigma(j) & \text{if } h(j) = i \\ 0 & \text{o.w.} \end{cases}$$

2a (4 pts) Show that Πx can be maintained at unit cost per update in the streaming model of computation, and that for every x the vector Πx can be computed in time proportional to the number of nonzeros in x.

2b (26 pts) Prove that if $m = C_2/\epsilon^2$ for a sufficiently large absolute constant $C_2 > 0$, then

$$(1-\epsilon)||x||_2^2 \leq ||\Pi x||_2^2 \leq (1+\epsilon)||\Pi x||_2^2$$

with probability at least 2/3 for every fixed $x \in \mathbb{R}^n$. *Hint: use Chebyshev's inequality on* $||\Pi x||_2^2$. *Computing the variance of* $||\Pi x||_2^2$ *is somewhat more involved than the variance calculation for the AMS sketch seen in class.*

3 (15 pts) For an unweighted graph $G = (V, E)$, we define its Laplacian matrix, $L \in \mathbb{R}^{V \times V}$, as follows: For any $u \in V$, $L_{u,u} = \text{degree}(u)$, and for any $u, v \in V$ let $L_{u,v} = L_{v,u} = -1$ if $e = \{u, v\} \in E$ and let $L_{u,v} = L_{v,u} = 0$ otherwise.

In this problem you will design a single pass streaming algorithm that processes the stream using $O(n \log n)$ bits of space and for any vector $x \in \mathbb{R}^V$ given at the end of the stream can approximate the quadratic form $x^\top Lx$, to within a factor 2, with success probability 0.9 (note that $x^\top$ is the transpose of vector x). In other words, given $x \in \mathbb{R}^V$ at the end of the stream, the algorithm must output an estimate $\eta > 0$ such that

$$\frac{1}{2}\eta \leq x^\top Lx \leq 2\eta.$$

3a (5 pts) Let $B \in \{-1, 0, 1\}^{\binom{n}{2} \times n}$ denote the edge incidence matrix of G, defined as follows. Rows of B are indexed by pairs $\{u, v\}$ of vertices in V. The row $\{u, v\}$ is zero if $\{u, v\} \notin E$, and otherwise contains exactly two nonzero entries: -1 at u and $+1$ at v (the signs can be chosen arbitrarily, as long as one is a plus and the other is a minus). Prove that

$$B^\top B = L,$$

where $B^\top$ is the transpose of matrix B.

3b (10 pts) Design a random matrix S with a constant number of rows such that $x^\top Lx$ can be approximated using $S \cdot B$, and show how this can be used to obtain the required streaming algorithm that estimates η.

4 **Facility location.** *(15 pts)* A famous Swiss chocolate factory has decided to ramp up production of chocolate, and would like to open $k > 0$ new locations from a short list of n candidate locations to optimally serve their clients (there are m clients, numbered 1 through m). If a subset $S \subseteq [n] = \{1, 2, \ldots, n\}$ of locations is open, for every $i = 1, \ldots, m$ the i-th client will go to the location that serves their favorite type of chocolate as specified by the client satisfaction matrix $A \in \mathbb{R}^{m \times n}$: the (i, j) entry of A specifies how much client i likes chocolate served by location j (the chocolate is excellent everywhere, so the entries of A are non-negative). Formally, we would like to find

$$\max_{S \subseteq [n], |S| \leq k} f(S),$$

where

$$f(S) = \sum_{i \in [m]} \max_{j \in S} A_{ij}$$

when $S \neq \emptyset$ and $f(\emptyset) = 0$. Give a polynomial time algorithm that achieves a $(1 - 1/e)$ approximation for this problem.

5 *(15 pts)* **Implementation.** The objective of this problem is to successfully solve the problem *Hypergraph cuts* on our online judge. In this problem you are given a 3-uniform hypergraph $G = (V, E)$ and your task is to find size of the the minimum cut in G, as well as the number of minimum cuts, using an extension of Karger's mincut algorithm. Recall that a subset $F \subseteq E$ is a cut in G if the hypergraph $G' = (V, E \setminus F)$ is disconnected, i.e. there exists a nonempty set $S \subset V$ of vertices such that all hyperedges in $E \setminus F$ are either contained in S or in $V \setminus S$. A minimum cut in G is a cut F that minimizes $|F|$ (the number of hyperedges in F).

You will find detailed instructions on how to do this on Moodle.