Homework 7 – Due Thursday, October 17, 2024

Page limit You can submit **at most** 2 pages per problem, even if the problem has multiple parts. If you submit a longer solution for some problem, only the first sheet of paper will be graded.

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the instructor if asked. You must also identify your collaborators and whether you gave help, received help, or worked something out together. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises Please practice on exercises in Chapter 4 of Mitzenmacher-Upfal.

Problems

1. (**Randomized Rounding and Randomized Response**)

   (a) (**Randomized Rounding**) This problem is similar to the Set Balancing problem discussed in class. We are given an $n \times n$ matrix $\mathbf{A}$ with entries in $\{0, 1\}$ and a real-valued vector $\mathbf{p}$ with n entries in $[0, 1]$. We would like to "round" the entries of $\mathbf{p}$ to obtain an n-bit vector $\mathbf{q}$ such that $\mathbf{Aq}$ is close to $\mathbf{Ap}$ in every component. In other words, we would like to ensure that $\|\mathbf{A}(\mathbf{p} - \mathbf{q})\|_\infty$ is small.

   To obtain $\mathbf{q}$, we can use the following approach, called *randomized rounding*: Each entry q_i of $\mathbf{q}$, for $i \in [n]$, is (independently) set to 1 with probability p_i and to 0 with probability $1 - p_i$. Derive a bound on $\|\mathbf{A}(\mathbf{p} - \mathbf{q})\|_\infty$ that is exceeded with probability at most $2/n$.

   (b) (**Randomized Response**) You are collecting sensitive data from n people. Each person i has a secret bit $x_i \in \{0, 1\}$. For example, this bit might represent whether each person already had COVID-19 or not. You would like to approximate the sum of the secret bits. (In our example, it would represent the number of people who had COVID-19.) To protect privacy of your respondents, you ask each of them to report X_i, which is set to x_i with probability $\frac{1}{2} + \epsilon$ and to $1 - x_i$ with probability $\frac{1}{2} - \epsilon$, where $\epsilon \in (0, \frac{1}{2})$ is a privacy parameter.

   You define new random variables Y_i, where Y_i is a function of X_i and ϵ, and return $Y = \sum_{i \in [n]} Y_i$ as your estimate for $\sum_{i \in [n]} x_i$.

   Explain how to define Y_i and derive upper bounds on the probability that the error of your estimate exceeds δn using (i) Chebyshev's inequality; (ii) Hoeffding bound.

   (Compute, but do not submit: What would δ be as a function of ϵ and n if you wanted constant probability of error (using asymptotic notation)? Is it different for the two bounds you obtained? What changes if you increase δ by a factor of $\sqrt{\ln n}$?)

   (c) (**Randomized Response on a Graph**) Now you use the same strategy to extract information about people's relationships. The input to the problem is an n-node graph, where vertices represent people, and an edge between two nodes represents that its endpoints engaged in some sensitive transaction. For each pair of nodes $e = \{u, v\}$, there is a secret bit x_e that indicates whether the edge e is present in the graph. Your goal is to estimate the number of triangles (i.e., 3-cycles) in the graph.

As before, you ask the participants to report X_e that is set to x_e with probability $\frac{1}{2}+\epsilon$ and to $1-x_e$ with probability $\frac{1}{2}-\epsilon$, where $\epsilon \in (0, \frac{1}{2})$ is a privacy parameter. You defined Y_e as in part (b). For each triple $\{u, v, w\}$ of nodes in the graph, you define a random variable $Z_{\{u,v,w\}} = Y_{\{u,v\}} \cdot Y_{\{v,w\}} \cdot Y_{\{u,w\}}$. Finally, you set Z to be the sum of the $\binom{n}{3}$ random variables $Z_{\{u,v,w\}}$.

Show that Z is an unbiased estimate of the number of triangles in the graph, that is, $\mathbb{E}[Z]$ is the number of triangles.

2. (**Randomized Routing on the Hypercube**) Recall the Randomized Routing Algorithm and its analysis from class. As part of the analysis, we stated a lemma that we did not have time to prove. In this problem, you will prove this lemma.

   (a) Consider each route in Phase 1 of the algorithm as a directed path from the source x to the designation z. Prove that once two routes separate, they do not rejoin.

   (b) Does part (a) imply that, for any two packets i and j, there is at most one node such that i and j are waiting in queue at that node at the same time step?

   (c) Consider any packet i. Let $p_i = (v_1, \ldots, v_k)$ be its path in phase 1. Let S be the set of packets (other than i) whose routes pass through at least one edge of p_i. Recall that the delay of a packet is the number of time steps it waits in queues (in Phase 1). Show that the delay of packet i is at most $|S|$.

   Hint: Use part (a).

   Guidelines: For each unit of delay that packet i encounters, we would like to "charge" one of the packets in S. We define the *lag* of a packet i' on the edge (v_j, v_{j+1}) as $t-j$, where t is the time step when i' traverses the edge (v_j, v_{j+1}). We say that a packet i' *leaves* p_i with lag ℓ if the lag of packet i' on the last edge of p_i it traverses is ℓ.

      i. Argue that if the delay of the packet i increases from ℓ to $\ell+1$ (for any integer ℓ), then there exists a packet i' from S that leaves p_i with lag ℓ. (You can charge i' for this unit of delay.)

      ii. Argue that you are charging each packet in S for at most one unit of delay and conclude that the delay of packet i is at most $|S|$.

3. **(Permutation Routing with Bit-Fixing in Random Order)** Recall the permutation routing problem on the n-dimensional hypercube we considered in class. Suppose n is even and recall that $N = 2^n$ is the number of nodes in the hypercube. In the *transpose permutation*, we want to route each packet with source $x_1 \ldots x_n$ to destination $x_{n/2+1} \ldots x_n x_1 \ldots x_{n/2}$.

   Before we studied the Randomized Routing Algorithm, Anatoly and Adrish[1] proposed to modify the bit-fixing algorithm, so that each packet chooses a random order of bits (independently from other packets) and then fixes bits in that order (instead of fixing them in the order from 1 to n). Show that this algorithm takes $2^{\Omega(n)}$ steps on the transpose permutation with high probability, following the guidelines in each part.

   (a) Consider packets that have exactly k bits $x_1, \ldots, x_{n/2}$ set to 1 and $x_{n/2+1} = \cdots = x_n = 0$, where k will be chosen later. What is the expected number of packets like that going through the node 0^n (i.e., with the all-zero label)?

[1] Your name could appear on the next homework! David asked how to achieve this. You can propose an interesting modification to an algorithm in class either during lecture or on Piazza.

(b) Use $(\frac{n}{k})^k \le \binom{n}{k} \le (\frac{en}{k})^k$ to give a lower bound on the expectation you computed and select a setting of k for which this bound is $2^{\Omega(n)}$. (You may ignore rounding issues, i.e., not worry about the fact that k has to be an integer.)

(c) Let B be your bound from the previous part. Use Chernoff-Hoeffding bounds to show that with high probability at least $B/2$ packets go through node 0^n.

(d) Complete the proof of the required lower bound on the number of steps. (Be careful: each *edge* can move one packet per time step, and so far we only argued about packets moving through a specific *node*.).