CS 537: *Randomness in Computing* | Prof. Sofya Raskhodnikova
Boston University | February 29, 2020

Homework 7 – Due Friday, March 6, 2020 at noon

Submit solutions to all problems on separate sheets. They will be graded by different people.

Page limit You can submit **at most** 1 sheet of paper 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. (**Using Chernoff/Hoeffding bounds**)

   (a) (**Amplification of the success probability**) You have a randomized algorithm $\mathcal{A}$ that outputs a real number and runs in time $T(n)$ on inputs of size n. It produces a good approximation (that is, a value in the range that you think is acceptable) to the problem you are trying to solve with probability at least $\frac{2}{3}$. Give an algorithm that, for a parameter $\delta \in (0,1)$, produces a good approximation (in the same sense as $\mathcal{A}$) with probability $\geq 1-\delta$ and runs in time $O(T(n)\log\frac{1}{\delta})$. (*Hint:* run the algorithm $\Theta(\log\frac{1}{\delta})$ times and output the median answer.)

   (b) (**Coronavirus resilience**) Suppose that an α fraction of the population is coronavirus-resilient. Let m be a natural number. Prove that if $s \geq \max\left\{\frac{2m}{\alpha}, \frac{8\ln(1/\delta)}{\alpha}\right\}$ then a sample of s people selected uniformly and independently at random with replacement will have at least m coronavirus-resilient people with probability at least $1-\delta$.

2. (**Random vectors**) Consider d-dimensional vectors, where d is a sufficiently large integer. Recall that two vectors $\mathbf{a}$ and $\mathbf{b}$ are orthogonal if their dot product $\mathbf{a}\cdot\mathbf{b} = \sum_{i\in[d]} a_i b_i$ is zero. For a real number $\epsilon > 0$, two vectors are *ϵ-close to being orthogonal* if their dot product is in the interval $[-\epsilon, \epsilon]$.

   (a) You pick two d-dimensional unit vectors independently at random by setting each coordinate uniformly and independently to $1/\sqrt{d}$ or $-1/\sqrt{d}$. Give the best upper bound you can on the probability that the two vectors are not $1/10$-close to orthogonal.

   (b) In d dimensions, at most d vectors can be pairwise orthogonal. However, exponentially many (in d) vectors can be close to pairwise orthogonal. Moreover, a random collection of exponentially many vectors is likely to be close to pairwise orthogonal. Find as large k as you can (as a function of d) such that k unit vectors chosen independently as specified in part (a) are pairwise $1/10$-close to being orthogonal with (at least) constant probability.

3. (**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 $|S|$.

Recall that, as part of our analysis of the Randomized Routing Algorithm, we defined a random variable Y_e to be the number of packets whose routes in Phase 1 use edge e of the hypercube. Calculate the expectation of Y_e using the following guidelines.

(d) Let L_i be the number of edges in the route of packet i. Argue that $\sum_e Y_e = \sum_i L_i$ for appropriately specified e and i. (Make sure you specify what e and i we are summing over.)

(e) Find the expectation of the summation in part (d).

(f) Use part (e) to derive $E[Y_e]$.