CS 537: *Randomness in Computing* — Prof. Sofya Raskhodnikova
Boston University — October 10, 2024

Homework 6 – Due Thursday, October 10, 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 2 pages 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. (**Improving guarantees of randomized algorithms**)

   (a) You have developed a randomized algorithm $\mathcal{A}$ that, on every input of length n, runs in time $O(n^2)$ and outputs either a correct answer for the problem you are trying to solve or "fail". You proved that, on every input, it returns "fail" with probability at most 0.99. Show how to modify algorithm $\mathcal{A}$ to get a new algorithm that always computes a correct answer and runs in expected time $O(n^2)$.

   (b) You have developed a randomized algorithm $\mathcal{B}$ that always solves your problem and, on every input of length n, runs in expected time $T(n)$. Show how to modify algorithm $\mathcal{B}$ to get a new algorithm that solves your problem with probability at least 0.95 and always runs in time $a \cdot T(n)$, for as small constant a as you can.

   Your friend managed to prove that the variance of the running time of $\mathcal{B}$ on input of length n is at most $\sqrt{n}$. How should you modify your solution above to obtain the best running time while still solving the problem with probability at least 0.95? Clarification: You answer for this question about modification does not need to be of the form $aT(n)$, where a is a constant. Also, if it is helpful, you may assume that $T(n) \geq n$: this should allow you to get a clear improvement in the running time when you use the bound on the variance.

   (c) You have developed a randomized algorithm $\mathcal{C}$ for computing a function f that, on every input x, returns the correct answer $f(x)$ with probability at least 0.7 and an incorrect answer with the remaining probability. To amplify the success probability, you do the following: you run your algorithm k times and output the answer that appears most frequently in the k runs (breaking ties arbitrarily). Let $t > 0$ be a parameter. Use a Chernoff-Hoeffding bound to find a value of k that ensures that the new algorithm makes a mistake in computing $f(x)$ with probability at most 2^{-t}.

2. (**Chernoff Bound for a die**) You have a fair 4-sided die with vertices marked with 0,1,2, and 3. Let random variable X denote the sum of n independent rolls of the die.

   (a) For all $\delta \in (0, 1)$, derive a Chernoff-type bound on $\Pr[X \geq (1 + \delta)\mathbb{E}[X]]$ using the method from Lecture 9 (i.e., applying Markov's inequality to the random variable e^{tX}). Instead of optimizing for the value of t, you may use $t = \ln(1 + \delta)$.

(b) Give an upper bound on $\Pr[X \geq 2n]$ using Markov's inequality, Chebyshev's inequality, Hoeffding bound (from Lecture 10), and your bound from part (a). Compare the four bounds you obtained.

3. (**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.

4*. **(Optional, no collaboration)** In this problem, you will analyze an algorithm for estimating the number of connected components in an undirected graph $G = (V, E)$ on n nodes within $\pm \epsilon n$, where $\epsilon \in (0, 1)$ is a parameter.

   (a) Let C be the number of connected components in G. For every node v, let n_v denote the number of nodes in the connected component of v. Prove that $C = \sum_{v \in V} \frac{1}{n_v}$.

   (b) For every node v, let $\hat{n}_v = \min(n_v, 2/\epsilon)$. Let $\hat{C} = \sum_{v \in V} \frac{1}{\hat{n}_v}$. Can $\hat{C}$ be smaller than C? Larger than C? By how much? (Give the best upper bound you can.)

   (c) Let s be a parameter. Define $\tilde{C}$ to be an estimate obtained as follows: *We sample s uniformly random nodes from G independently with replacement. For each sampled node v, we compute $\hat{n}_v$ by doing a BFS from v until we visit at most $2/\epsilon$ nodes. We compute the average of values $\frac{1}{\hat{n}_v}$ over all sampled nodes and set $\tilde{C}$ to be n times the average.*
   Use a Chernoff-Hoeffding bound to find the (asymptotically) smallest value of s for which

   $$\Pr[|\tilde{C} - \hat{C}| \geq \epsilon n/2] \leq 1/3.$$

   (d) Argue that, with probability at least $2/3$, the estimate $\tilde{C}$ approximates the number of connected components in G within $\pm \epsilon n$.

   (e) Suppose G has degree at most d. Assuming BFS takes constant time per node put in the BFS queue, how long does the procedure above take (asymptotically), with the setting of s that you found?