CS 537: *Randomness in Computing* — Prof. Sofya Raskhodnikova
Boston University — February 27, 2020

Homework 6 – Due Friday, February 28, 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. (**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 manages 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?

   (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. (**Exercise 4.10 from MU**) A casino is testing a new class of simple slot machines. Each game, the player puts in \$1, and the slot machine is supposed to return either \$3 to the player with probability 4/25, \$100 with probability 1/200, or nothing with all the remaining probability. Each game is supposed to be independent of other games.

   The casino has been surprised to find in testing that the machines have lost \$10,000 over the first million games. Your task is to come up with an upper bound on the probability of this event, assuming that their machines are working as specified.

(a) Use Theorem 4.12 (Hoeffding Bound) from the MU book to give an upper bound.

In the rest of the problem, you will derive a specialized Chernoff bound for this problem to get a better upper bound.

(b) Let the random variable X denote the *net loss* to the casino over the first million games. Derive an expression for $E[e^{tX}]$, where t is an arbitrary real number.

(c) Derive from first principles a Chernoff bound for the probability $\Pr[X \geq 10,000]$.

Hint: Follow the proof of the Chernoff bound in class, by applying Markov's inequality to the random variable e^{tX}. Use the value $t = 0.0006$ in your bound.)

3. (**Concentration for the running time of Randomized Quicksort**) MU Exercise 4.21.

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

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

(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) If G has degree at most d, how many nodes does the procedure above visit, with the setting of s that you found?