CS 537: *Randomness in Computing* — Prof. Sofya Raskhodnikova
Boston University — October 1, 2024

Homework 5 – Due Thursday, October 3, 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 3 of Mitzenmacher-Upfal.

Problems

1. (**MaxCut**) In the problem MaxCut, we are given an undirected graph $G = (V, E)$ and asked to find a cut of maximum size in G. (Recall that a cut in G is a partition of the vertex set V into two parts; the size of the cut is the number of edges with one endpoint in each part of the partition. Assume that G is simple, that is, it has no loops and no multiple edges.) In contrast to the seemingly very similar problem MinCut discussed in class (recall Karger's algorithm), MaxCut is a famous NP-hard problem, so we do not expect to find an efficient algorithm that solves it exactly. Here is a simple linear-time randomized algorithm that gives a pretty good approximation:

   - Randomly and independently color each vertex $v \in V$ red or blue with probability $1/2$.
   - Output the cut defined by the red/blue partition of the vertices.

   (a) Let random variable X denote the size of the cut output by the algorithm. Compute $\mathbb{E}[X]$ as a function of the number of edges in G, and deduce that $\mathbb{E}[X] \geq OPT/2$, where OPT is the size of a maximum cut in G.

   Hint: Write X as a sum of indicator random variables.

   (b) Let p denote the probability that the cut output by the algorithm has size at least $0.49\, OPT$. Show that $p \geq 1/51$.

   Hint: Applying Markov's inequality to X will not work here. Try applying Markov's inequality to a different random variable.

   (c) Now compute the variance $\mathrm{Var}[X]$.

   Hint: Again write X as the sum of indicators, as in part (a).

   (d) Let p be the probability defined in part (b). Use Chebyshev's inequality together with part (c) to show that $p = 1 - O(1/|E|)$.

   (Note how Chebyshev's inequality gives us a better bound here than Markov's.)

2. (**Random counter, 20 points, double the usual page limit**) You want to create a counter that stores the number of times the door to your lab has been opened. Assume that the door will be opened at most m times. You initialize your counter to 0.

(a) (**Warmup**) Suppose you increment your counter by 1 deterministically: every time the door is opened. How many bits do you need to represent your counter?

Now you want to study a different (randomized) implementation of the counter.

(b) Let X be the current value of the counter. Suppose you increment it by 1 randomly: every time the door is opened, you do it with probability 2^{-X}. When you want an estimate of the number of times the door has been opened, you compute $2^X - 1$.

Let X_i be the value of your counter when the door has been opened i times. Use the compact form of the law of total expectation to compute the expectation of 2^{X_i}.

(c) How many bits do you need to represent your counter, assuming that 2^X never exceeds 10 times its expectation (for all steps in the process)? Use asymptotic notation to express your answer.

(d) Compute the variance of 2^{X_i}.

(e) Let $\tilde{m}$ be the value of $2^X - 1$ after the door has been opened m times. Fix a parameter $\epsilon > 0$. We say that $\tilde{m}$ is a $(1 \pm \epsilon)$-approximation of m if

$$(1-\epsilon)m \leq \tilde{m} \leq (1+\epsilon)m.$$

Use Chebyshev's inequality to give an upper bound in terms of ϵ on the probability that $\tilde{m}$ is NOT a $(1 \pm \epsilon)$-approximation of m.

(f) To decrease the error probability, you decide to keep t independent random counters. That is, every time the door opens, each counter X_j for $j \in [t]$ is incremented with probability 2^{-X_j}, and the random coins used for different counters are mutually independent. Your new estimate is the average of the estimates you get from the counters: $\tilde{m} = \frac{1}{t}\sum_{j\in[t]}(2^{X_j} - 1)$. What is the expectation and variance of the new estimate?

(g) Redo part (e) for the new estimate to get a bound in terms of ϵ and t. What should t be set to as a function of ϵ to ensure that $\tilde{m}$ is a $(1 \pm \epsilon)$-approximation of m with probability at least 99%?

3. (**Generalization of Randomized Median Algorithm**) In this problem, you are asked to generalize the randomized median-finding algorithm from class (see Lecture 8 and also Section 3.5 of the MU book), so that it finds an element of rank k (that is, the kth smallest element) in an array of n distinct elements, for any given $k \in [4n^{3/4}, n - 4n^{3/4}]$. You may ignore rounding issues in your algorithm and analysis.

   (a) Explain how to modify lines $3, 4, 6$ and 8 of the algorithm on the slide 11 from Lecture 8.

   (b) Analyze the running time of the modified algorithm.

   (c) We will follow the same analysis outline as in class (and in the book). Change the definitions of events $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_{3,1}$ and $\mathcal{E}_{3,2}$, so that they apply for general k.

   (d) Write $\Pr[\mathcal{E}_1]$ and $\Pr[\mathcal{E}_{3,1}]$ as probability expressions involving the tail of a suitable binomial random variable.

   Do *not* repeat the rest of the analysis (which is essentially the same as in the case of finding the median).