1. (**A Hiring Strategy.**) You need a new staff assistant and you have n people to interview. You want to hire the best candidate for the position. You interview the candidate one by one. After you interview the kth candidate, you either offer the candidate the job before the next interview or you forever lose the chance to hire that candidate. We suppose the candidates are interviewed in a random order (all $n!$ possible orderings being equally likely).

We consider the following strategy. First, interview m candidates but reject them all. After that, hire the first candidate who is better than the first m candidates.

(a) Conditioned upon the best candidate being the jth candidate, show that the probability that we hire the best candidate is given by:

$$\begin{cases} \frac{m}{j-1} & \text{if } j > m \\ 0 & \text{otherwise} \end{cases}$$

(b) Compute the probability that we hire the best candidate (this should be an expression in terms of m and n). In addition, show that if we set $m = n/e$, then the probability that we hire the best candidate is approximately $1/e$. [HINT: Use the fact that $1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} \approx \ln n$.]

2. **(Parameter Estimation.)** Suppose we have population of $n = 20$ million people and we want to estimate the fraction p of Republicans (assume $p > 0.01$). Here's one way to do it:

> Interview a sample of s people chosen independently and uniformly at random from the population, with replacement. Output the fraction of them who are Republicans.

Suppose we want the algorithm to output a value in the interval $(p - \delta, p + \delta)$ (where $\delta \leq 0.01$) with probability at least $1 - \epsilon$. Then, how large should s be? Compute an upper bound on s (as a function of n, δ and ϵ) using both the Chebyshev's inequality and the Chernoff bound. Then, compare the bounds for the specific values $n = 20$ million, $\delta = 0.01$ and $\epsilon = 0.01$.

3. **(Much Ado About Max-Cuts.)** In the problem MAXCUT, we are given an undirected graph $G = (V, E)$ and asked to find a cut of *maximum* size in G. In contrast to the seemingly very similar problem MINCUT discussed in class (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 very 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 $\frac{1}{2}$ each;
- output the cut defined by the red/blue partition of vertices.

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

(b) Let p denote the probability that the cut output by the algorithm has size at least 0.49OPT. 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 r.v.]

(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 as 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 much sharper bound here than Markov.]

(e) How would you modify the algorithm so that it *always* finds a cut of size at least 0.49OPT but has only *expected* linear running time?