PRINCETON UNIV FALL '22		COS 521: ADVANCED ALGORITHMS
	Homework 3	
Out: *Oct 11*		Due: *Oct 31*

Instructions:

- Upload your solutions (to the non-extra-credit) to each problem as a **single PDF** file (one PDF total) to Gradescope. Please make sure you are uploading the correct PDF! Please anonymize your submission (i.e., do not list your name in the PDF), but if you forget, it's OK.

- If you choose to do extra credit, upload your solution to the extra credits as a single separate PDF file to Gradescope. Please again anonymize your submission.

- You may collaborate with any classmates, textbooks, the Internet, etc. Please upload a brief "collaboration statement" listing any collaborators as a separate PDF on Gradescope (if you forget, it's OK). But always **write up your solutions individually**.

- For each problem, you should have a solid writeup that clearly states key, concrete lemmas towards your full solution (and then you should prove those lemmas). A reader should be able to read any definitions, plus your lemma statements, and quickly conclude from these that your outline is correct. This is the most important part of your writeup, and the precise statements of your lemmas should tie together in a correct logical chain.

- A reader should also be able to verify the proof of each lemma statement in your outline, although it is OK to skip proofs that are clear without justification (and it is OK to skip tedious calculations). Expect to learn throughout the semester what typically counts as 'clear'.

- You can use the style of Lecture Notes and Staff Solutions as a guide. These tend to break down proofs into roughly the same style of concrete lemmas you are expected to do on homeworks. However, they also tend to prove each lemma in slightly more detail than is necessary on PSets (for example, they give proofs of some small claims/observations that would be OK to state without proof on a PSet).

- Each problem is worth twenty points (even those with multiple subparts), unless explicitly stated otherwise.

Problems:

§1 We say a random variable Z is *subgamma* with parameters (σ^2, B), if

$$\mathbb{E}\left[e^{\lambda(Z-\mathbb{E}[Z])}\right] \le e^{\lambda^2\sigma^2/2},$$

for all $|\lambda| \le B$.

(a) Let $Z_1, \ldots, Z_m$ be *independent* subgamma random variables with parameters (σ_i^2, B_i) respectively for $i \in [m]$. Prove that $\sum_{i\in[m]} Z_i$ is subgamma with parameters $(\sum_{i\in[m]} \sigma_i^2, \min_{i\in[m]} B_i)$.

(b) Show that if Z is subgamma with parameters (σ^2, B), then for any $t > 0$, both $\Pr[Z - \mathbb{E}[Z] > t]$ and $\Pr[Z - \mathbb{E}[Z] < -t]$ are at most $\max\left\{e^{-\frac{t^2}{2\sigma^2}}, e^{-\frac{tB}{2}}\right\}$.

(c) Let Z be a geometric random variable such that $\Pr[Z = k] = p \cdot (1-p)^{k-1}$ for all integers $k \ge 1$. Prove that Z is subgamma with parameters $(2/p^2, p/2)$.

Hint: The following two inequalities may be useful: $e^x \ge 1 + x$ for all $x \in \mathbb{R}$ and $1/(1-x) \le e^{x+x^2}$ for $|x| \le 1/2$.

§2 In this problem, we will analyze the *Morris counter* briefly mentioned in the lecture, and prove that it solves approximate counting using $O(\log\log N + \log(1/\varepsilon) + \log\log(1/\delta))$ bits of space.

Recall that the approximate counting problem asks us to maintain a counter n (initialized to 0) up to N, supporting

- `inc()`: $n \leftarrow n + 1$;
- `query()`: output $\tilde{n}$ such that $\Pr[|\tilde{n} - n| > \varepsilon n] < \delta$.

A Morris counter has a parameter $\alpha > 0$. It maintains a variable X, initialized to 0. Each time `inc()` is called, X is incremented to $X + 1$ with probability $(1+\alpha)^{-X}$. When `query()` is called, it returns $\tilde{n}(X) = ((1+\alpha)^X - 1)/\alpha$.

(a) Let Y_k be the random variable denoting the number of `inc()` calls needed to increment X from $k-1$ to k. Derive the probability distribution of each Y_k, and prove that it is subgamma for some parameters (σ_k^2, B_k).

(b) When $\alpha n > C$ for a sufficiently large constant C, prove that after n `inc()` calls, $\tilde{n}(X) < (1-\varepsilon)n$ with probability at most $e^{-\Omega(\varepsilon^2/\alpha)}$.

We can prove a similar upper bound on the probability that $\tilde{n}(X) > (1+\varepsilon)n$, and you may use this bound without a proof in (c).

(c) Given N, ε, δ, choose the right parameters T and α, and prove that the Morris counter, together with an exact counter when $n \le T$ using $O(\log T)$ bits, solves approximate counting with $O(\log\log N + \log(1/\varepsilon) + \log\log(1/\delta))$ bits.

§3 The ℓ_1 distance between vectors $x, y \in \mathbb{R}^d$ is defined as $\|x - y\|_1 = \sum_{i=1}^d |x_i - y_i|$. Consider the Johnson-Lindenstrauss dimensionality reduction method described in lecture: $x \to \Pi x$ where each entry in $\Pi \in \mathbb{R}^{m \times d}$ equals

$$\Pi_{ij} = c \cdot g_{ij},$$

for some fixed scaling factor c and $g_{ij} \sim \mathcal{N}(0, 1)$. Describe an example (i.e., a set of points in $\mathbb{R}^d$) which shows that, for any choice of c, this method does *not* preserve ℓ_1 distances, even within a factor of 2. You may pick a single d for your example.

Hint: You may want to use the fact that this choice of Π preserves ℓ_2 distances.

§4 Given a data matrix $X \in \mathbb{R}^{n \times d}$ with n rows (data points) $x_1, \ldots, x_n \in \mathbb{R}^d$, the k-*means clustering problem* asks us to find a partition of our points into k disjoint sets (clusters) $\mathcal{C}_1, \ldots \mathcal{C}_k \subseteq \{1, \ldots, n\}$ with $\bigcup_{j=1}^k \mathcal{C}_j = \{1, \ldots, n\}$.

Let $c_j = \frac{1}{|\mathcal{C}_j|} \sum_{i \in \mathcal{C}_j} x_i$ be the centroid of cluster j. We want to choose our clusters to minimize the sum of squared distances from every point to its cluster centroid. I.e. we want to choose $\mathcal{C}_1, \ldots \mathcal{C}_k$ to minimize:

$$f_X(\mathcal{C}_1, \ldots \mathcal{C}_k) = \sum_{j=1}^k \sum_{i \in \mathcal{C}_j} \|c_j - x_i\|_2^2.$$

There are a number of algorithms for solving the k-means clustering problem. They typically run more slowly for higher dimensional data points, i.e. when d is larger. In this problem we consider what sort of approximation we can achieve if we instead solve the problem using dimensionality reduced vectors in place of $x_1, \ldots, x_n$.

Let $OPT_X = \min_{\mathcal{C}_1, \ldots \mathcal{C}_k} f_X(\mathcal{C}_1, \ldots \mathcal{C}_k)$.

Suppose that Π is a Johnson-Lindenstrauss map into $s = O(\log n/\epsilon^2)$ dimensions and that we select the optimal set of clusters for $\Pi x_1, \ldots, \Pi x_n$. Call these clusters them $\tilde{\mathcal{C}}_1, \ldots \tilde{\mathcal{C}}_k$. Show that they obtain objective value $f_X(\tilde{\mathcal{C}}_1, \ldots \tilde{\mathcal{C}}_k) \leq (1 + \epsilon) OPT_X$, with high probability.

Hint: reformulate the objective function to only involve ℓ_2 distances between data points.

§5 In class, we saw that for any graph on n vertices, there exists a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges. Prove that there exists some graph G, for which there is no $(k-1)$-spanner with $O(n^{1+1/k})$ edges.

Hint: Consider a random graph G where every pair (u, v) has an edge with probability $p \approx 1/n^{1-1/k}$ independently.

Extra Credit:

§1 (Extra credit) Let $\Pi \in \mathbb{R}^{d \times m}$ be a sparse random matrix such that every column has exactly s random non-zero entries, and every non-zero entry is a random $\pm 1/\sqrt{s}$. In class, we stated (without a proof) that by setting $d = O(\varepsilon^{-2}\log(1/\delta))$ and $s = O(\varepsilon d)$, we have that $\forall x \in \mathbb{R}^m$,

$$\Pr_{\Pi}[\|\Pi\|_2^2 = (1 \pm \varepsilon)\|x\|_2^2] > 1 - \delta.$$

Prove that if we put a random Guassian $\mathcal{N}(0, 1/s)$ in every non-zero entry instead of $\pm 1/\sqrt{s}$, we will not have the same guarantee.