Princeton Univ Fall '25	COS 521: Advanced Algorithms
Homework 3	
Out: *Oct 7*	Due: *Oct 26*

Instructions:

- Upload your non-extra solutions to Gradescope in a single PDF file, and mark your solution to each problem. 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 A k-sparse vector is any vector with at most k nonzero entries. Let $\mathcal{S}_k$ be the set of all k-sparse vectors in $\mathbb{R}^d$. Show that, if Π is chosen to be a Johnson-Lindenstrauss embedding matrix (e.g. a scaled random Gaussian matrix) with $s = O(\frac{k \log d}{\epsilon^2})$ rows then, with high probability,

$$(1-\epsilon)\|\Pi x\|_2 \leq \|x\|_2 \leq (1+\epsilon)\|\Pi x\|_2$$

for all $x \in \mathcal{S}_k$, simultaneously.

Hint: You will want to use some result from the JL lecture as a black-box.

§2 (a) Let $m \geq 1$ be an integer, prove that there can be at most $2^{O(m)}$ points in $\mathbb{R}^m$ such that the distance between *every* pair of points is between 1 and 3.

(b) For any $n \geq 1$, construct a set of n points in $\mathbb{R}^n$ such that the distance between *every* pair of points is equal to 2.

(c) Prove that the new dimension in the JL lemma is optimal up to a constant factor when $\varepsilon = 0.1$, i.e., in general, we cannot hope to map an arbitrary set of n points in a high dimensional space to $\mathbb{R}^{c \log n}$ while the pairwise distances are preserved up to a 1 ± 0.1 factor, when $c > 0$ is a sufficiently small constant.

§3 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 $\tilde{\mathcal{C}}_1, \ldots \tilde{\mathcal{C}}_k$. 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.

§4 Recall the max-flow problem from undergraduate algorithms: for a directed graph $G(V, E)$ with non-negative capacities c_e for every $e \in E$ and two special vertices s (source, with no incoming edges) and t (sink, with no outgoing edges), a *flow* in G is an assignment $f : E \to \mathbb{R}_{\geq 0}$ such that $f(e) \leq c_e$ for every edge and for every vertex $v \in V \setminus \{s, t\}$, the total incoming flow $\sum_{(u,v)\in E} f((u, v))$ equals the total outgoing flow $\sum_{(v,w)\in E} f((v, w))$. The task is to find a maximum flow f i.e., a flow f such that $\sum_{(s,u)\in E} f((s, u))$ is maximized.

(a) Show that the following LP is a valid formulation for computing the value of the maximum flow in G. There is a variable $f((u, v))$ for all $(u, v) \in E$.

$$\begin{aligned} \max & \sum_u f((u,t)) \\ & \forall e = (u,v) \in E,\ f((u,v)) \leq c_e \\ & \forall v \notin \{s,t\},\ \sum_{(u,v)\in E} f((u,v)) = \sum_{(v,w)\in E} f((v,w)) \\ & \forall e \in E,\ f(e) \geq 0 \end{aligned} \tag{1}$$

(b) Write the dual for the LP (1). Show that this dual LP computes the minimum *fractional* s-t cut in G. A fractional cut places each node v at some point y_v on the unit interval $[0, 1]$, with s placed at 0 and t placed at 1. The value of the fractional cut is $\sum_{(u,v)\in E(G)} c_e \cdot \max\{0, y_v - y_u\}$ (where c_e is the weight of edge e). Observe that if instead each $y_v \in \{0, 1\}$, that this is simply an s-t cut. Use strong LP duality to conclude the *fractional* max-flow min-cut theorem. That is, if the max-flow is C, there exists a fractional s-t cut of value C, and no fractional s-t cut of value $< C$.

(c) Devise a rounding scheme that takes as input a fractional min-cut of value C and outputs a true (deterministic) min-cut of value C. (Hint: there is a simple rounding scheme that works, but it is not a rounding scheme we have already seen in class. You might want to first construct a randomized min-cut.) Conclude the max-flow min-cut theorem.

§5 (Firehouse location) Suppose we model a city as an m-point finite metric space with $d(x, y)$ denoting the distance between points x, y. These $\binom{m}{2}$ distances (which satisfy triangle inequality) are given as part of the input. The city has n houses located at points $v_1, v_2, \ldots, v_n$ in this metric space. The city wishes to build k firehouses and asks you to help find the best locations $c_1, c_2, \ldots, c_k$ for them, which can be located at any of the m points in the city. The *happiness* of a town resident with the final locations depends upon his distance from the closest firehouse. So you decide to minimize the cost function $\sum_{i=1}^{n} d(v_i, u_i)$ where $u_i \in \{c_1, c_2, \ldots, c_k\}$ is the firehouse closest to v_i. Describe an LP-rounding-based algorithm that runs in poly(m) time and solves this problem approximately. If OPT is the optimum cost of a solution with k firehouses, your solution is allowed to use $O(k \log n)$ firehouses and have cost at most OPT.[1]

[1]The term for an approximation guarantee like this is *resource augmentation* — the solution is as good as the optimum, but it requires additional firehouses.

Specifically, you should design an algorithm which runs in polynomial time, and uses $O(k \log n)$ firehouses in expectation, and also has cost at most OPT in expectation.[2]

Hint: "Oversample" to preserve the cost of the solution, while increasing the expected number of firehouses

Extra Credit:

§1 (Extra credit) Prove that n vectors in $\mathbb{R}^m$ cannot always be mapped to a $c\epsilon^{-2}$-dimensional space for a sufficiently small constant c, while preserving the pairwise distances within $1 \pm \epsilon$ factors.

[2]You may want to briefly think about how to modify your solution to run in expected polynomial time and use $O(k \log n)$ firehouses with probability one, or how to run in expected polynomial time and guarantee a solution with cost $(1 + \varepsilon)$OPT with probability one (or both). But you do not need to write this for full credit.