Problem 1: Recall that in the indexing problem from class, Alice is given $x \in \{0, 1\}^n$ and Bob is given $j \in [n]$. Alice then sends a single message to Bob from which Bob must recover $x_j$ with probability at least $1 - \delta$. We showed in class that any such message Alice sends must be of length at least $(1 - H_2(\delta))n$ where $H_2(\delta) = -\delta \log_2(\delta) - (1 - \delta) \log_2(1 - \delta)$. We also introduced the communication problem gap hamming in which Alice and Bob receive $x, y \in \{0, 1\}^N$, respectively, and are promised that either $\Delta(x, y) > N/2 + \sqrt{N}$ or $\Delta(x, y) < N/2 - \sqrt{N}$ and must decide which is the case. Here $\Delta(x, y)$ is the Hamming distance between $x, y$, i.e. the number of coordinates where they differ. By a reduction, for some $c > 0$ any one-way communication protocol for gap hamming with $11/12$th success probability for some $N = cn$ can be used to give a protocol for indexing with success probability $2/3$, where Alice's input has length $n$. Thus gap hamming with $11/12$th success probability requires one way communication of $\Omega(N)$ bits.

(a) (2 points) Use the gap hamming lower bound to show that any $(1 + \varepsilon)$-approximation to the $\ell_2$-norm in turnstile streams requires $\Omega(1/\varepsilon^2)$ bits of space as long as $1/\varepsilon^2 < n$.

(b) (3 points) Let us make Bob's life easier by giving him not only $j \in [n]$ but also $x_1, \ldots, x_{j-1}$. Show that, even giving Bob this extra information, Alice's message still needs to be of length at least $(1 - H_2(\delta))n$ to solve indexing with $\delta$ failure probability.

(c) (5 points) Use part (b) to show that, in fact, any $(1 + \varepsilon)$-approximation to the $\ell_2$-norm, even in strict turnstile streams, requires $\Omega((\log(\varepsilon^2 n)/\varepsilon^2))$ bits of space as long as $\log(\varepsilon^2 n)/\varepsilon^2 < n$. Recall that in strict turnstile streams, some vector $x \in \mathbb{R}^n$ receives coordinate-wise updates (either negative or positive), but we are promised that $x_i \geq 0$ for all $i$ at all points in the stream.

Problem 2: In the $k$-means clustering problem the input consists of $x_1, \ldots, x_N \in \mathbb{R}^n$ and a positive integer $k$, and the goal is to output some partition $\mathcal{P}$ of $[n]$ into $k$ disjoint subsets $P_1, \ldots, P_k$ as well as some $y = (y_1, \ldots, y_k) \in (\mathbb{R}^n)^k$ (the $y_i$ need not be equal to any of the $x_i$ and can be chosen arbitrarily) so as to minimize the cost function

$$
cost_{\mathcal{P}, y}(x_1, \ldots, x_N) = \sum_{j=1}^{k} \sum_{i \in P_j} \|x_i - y_j\|_2^2.
$$
That is, the $x_i$ are clustered into $k$ clusters according to $\mathcal{P}$, and the cost of a given clustering is the sum of squared Euclidean distances to the cluster centers (the $y_j$'s).

Unfortunately finding the optimal clustering for $k$-means is NP-hard, however efficient approximation algorithms do exist which find a clusterings that are close to optimal.

(a) (5 points) Given a partition $\mathcal{P}$ of $[n]$, show that the optimal $y$ vector to choose for that given $\mathcal{P}$ is the one where, for the $P_j$ of positive size, $y_j = \left(1/|P_j|\right) \cdot \sum_{i\in P_j} x_i$. Thus we can restrict our attention to just optimizing over $\mathcal{P}$.

(b) (5 points) Show that for any $0 < \varepsilon < 1/2$ there is a linear map $\Pi \in \mathbb{R}^{m \times n}$ for $m = O(\varepsilon^{-2} \log N)$ such that for all partitions $\mathcal{P}$ simultaneously,

$$ (1 - \varepsilon) \cdot \text{cost}(x_1, \ldots, x_N) \leq \text{cost}(\Pi x_1, \ldots, \Pi x_N) \leq (1 + \varepsilon) \cdot \text{cost}(x_1, \ldots, x_N), $$

and where $\Pi$ can be found efficiently with a randomized algorithm that has small failure probability. Thus, if one does not mind worsening the quality of solution found by a factor $1 + \varepsilon$, without loss of generality one can assume the input vectors $x_1, \ldots, x_N \in \mathbb{R}^n$ are in dimension $n = O(\varepsilon^{-2} \log N)$. 

2