Problem 1: (10 points) In class we claimed INDEX\textsubscript{cn} reduces to GAPHAM\textsubscript{n} for some constant $c$. In the former problem, Alice receives $x \in \{0, 1\}^c$ and Bob receives $j \in [n]$, and the goal is to output $x_j$. We showed $R_{1/3}^{\text{pub}}(\text{INDEX}_{cn}) = \Omega(n)$. Recall $R_p^{\text{pub}}(f)$ denotes the cost of the cheapest one-way communication protocol to compute $f$ with failure probability at most $p$ when Alice speaks first and the two players Alice and Bob share a common (public) random string. In GAPHAM\textsubscript{n}, Alice and Bob are given $x$ and $y$ respectively, each in $\{0, 1\}^n$. They are promised that either $\Delta(x, y) \leq n/2 - \sqrt{n}$, or $\Delta(x, y) > n/2 - \sqrt{n}$, and they must decide which. Here $\Delta$ denotes Hamming distance, i.e. number of coordinates in which the two vectors differ. The goal of this problem is to show, via reduction from INDEX\textsubscript{cn}, $R_{1/100}^{\text{pub}}(\text{GAPHAM}_{n}) = \Omega(n)$.

Consider the following approach to reduce from INDEX\textsubscript{cn}, in which Alice receives $x$ and Bob receives $j$. The two players break up their common random string into pieces $r_1, \ldots, r_n$, each in $\{0, 1\}^c$. Let $S = \{i : x_i = 1\}$. Alice creates a vector $y \in \{0, 1\}^n$ whose $k$th entry is the majority bit amongst the $r_i$ for $i \in S$ (if there is a tie, she chooses $y_k$ as a uniformly random bit). Bob creates a vector $z \in \{0, 1\}^n$ with $z_k = r_j^k$. Show that for some constant $c$, this transformation can be used as part of a reduction to show (1).

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

$$\text{cost}_{\mathcal{P}, y}(x_1, \ldots, x_N) = \sum_{j=1}^{k} \sum_{i \in P_j} ||x_i - y_j||_2^2.$$  

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, $y_j = (1/|P_j|) \cdot \sum_{i \in P_j} x_i$. 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 $n = O(\varepsilon^{-2} \log N)$. 