1. (40 points) **(Fun with Metrics)** In this exercise, you will learn how to use metric embeddings to design algorithms.

   (a) (10 points) Show how to solve the k-median problem on line metrics optimally. That is, given n clients C and m choices for placing the centers F, and a line metric d_L on $C \cup F$, find a subset $S \subseteq F$ such that $|S| = k$ and the $cost(C, S)$ is minimized. Recall that $cost(C, S) = \sum_{j \in C} d_L(j, S)$, and $d_L(j, S) = \min_{i \in S} d_L(j, i)$.

Solution:

Proof. We try to solve this problem optimally in polynomial time using dynamic programming.
First we sort the vertices on the line in $O(n \log n)$. Note we can do that as C and F can be put on a line according to the line metric. Hence there is a well defined notion of an order among points. Note that this is the crucial factor behind why the dynamic programming solution works.
Let $dp[i][j]$ be the minimum cost function when the number of medians chosen are j in number and the clients considered are only the first i clients.
Next we consider another array $dp2[i'][i]$ that has the minimum cost function when the number of clients to be assigned are between i' and i (both included) and number of medians to be chosen are 1. The total number of states in this dp are $\binom{n}{2}$. And to fill one state we require $|F|$ computations via brute force. Hence to fill this array requires $O(n^2|F|)$ complexity.
We now try to fill the first dp array through recursion and using the second dp array. Note that the base case is when $j = 1$, $\forall i$ from 1 to n. Using second dp array we have,

$$dp[i][1] = dp2[1][i]$$

The recursive relation in dp is as follows.

$$dp[i][j] = min_{1 \leq i' < i}\{dp[i'][j-1] + dp2[i'+1][i]\}$$

This recursive relation arises from the act that there must be some points in the end that belong to one median. This group of points can range from 1 to i. And we try to find the one that minimizes the objective function. Where the solution of the part that comes from a median in the end is found by using the second dp array.
Hence we can fill all the dp states using this recursive relation. The number of states in this dp array are n*k. And to fill each state we have to do a computation of $O(n)$. Hence the complexity here is $O(n^2k)$.
In the end the answer will be given by $dp[n][k]$ where k is required medians to be chosen. The complexity is $O(n^2k + n^2|F|)$. Which is equal to $O(n^2|F|)$. □

   (b) (10 points) Now assume the following (fake) theorem:

Theorem 1. *Given any metric (X, d) where X is a set of n points, and $d(\cdot)$ is a general metric distance function, we can efficiently find an embedding into a line metric d_L such that for all $u \neq v$, $d(u,v) \leq d_L(u,v) \leq O(\log n)d(u,v)$.*

Using the fake theorem above, devise a $O(\log n)$-approximation to k-median on general metrics. (Hint: consider the modified instance $\mathcal{I}' = (C \cup F, d_L)$ and solve k-median on d_L, and output the same solution for the original instance.)

Solution:

Proof. The part above gives us an optimal solution on the k median problem when the distance metric is a line metric. Let us solve the given problem on this modified line metric. Let the instance be $I' = (C \cup F, d_L)$. Let the solution of the k medians picked be the set Opt_{d_L}. We use this set as the k medians set in our original problem and claim that this is an $O(\log n)$ approximation. Claim of the proof is as follows

$$\begin{aligned}
cost_{d_L}(C, Opt_{d_L}) &= \sum_{j \in C} d_L(j, Opt_{d_L}),\ d_L(j, Opt_{d_L}) = \min_{i \in Opt_{d_L}} d_L(j,i)\\
cost(Alg) &= \sum_{j \in C} d(j, Opt_{d_L}),\ d(j, Opt_{d_L}) = \min_{i \in Opt_{d_L}} d(j,i)\\
cost(Opt) &= \sum_{j \in C} d(j, Opt),\ d(j, Opt) = \min_{i \in Opt} d(j,i)\\
&\text{Let, } d(j, Opt) = d(j, i_{Opt}),\ d_L(j, Opt_{d_L}) = d_L(j, i'')\\
d_L(j, i'') &\leq d_L(j, i_{Opt}), \text{ As i'' is the minimum}\\
d_L(j, i_{Opt}) &\leq O(\log n) d(j, i_{Opt}), \text{ From fake theorem}\\
d_L(j, i'') \leq d_L(j, i_{Opt}) &\leq O(\log n) d(j, i_{Opt})\\
d_L(j, Opt_{d_L}) &\leq O(\log n) d(j, Opt)\\
\sum_{j \in C} d_L(j, Opt_{d_L}) &\leq \sum_{j \in C} O(\log n) d(j, Opt)\\
cost_{d_L}(C, Opt_{d_L}) &\leq O(\log n) cost(Opt)
\end{aligned}$$

Also,

$$\begin{aligned}
&\text{Let, } d(j, Opt_{d_L}) = d(j, i'),\ d_L(j, Opt_{d_L}) = d_L(j, i'')\\
d(j, i') &\leq d(j, i''), \text{ As i' is the minimum}\\
d(j, i'') &\leq d_L(j, i''), \text{ From fake theorem}\\
d(j, Opt_{d_L}) &\leq d_L(j, Opt_{d_L})\\
\sum_{j \in C} d(j, Opt_{d_L}) &\leq \sum_{j \in C} d_L(j, Opt_{d_L})\\
cost(Alg) &\leq cost_{d_L}(C, Opt_{d_L})\\
\Rightarrow cost(Alg) &\leq O(\log n) cost(Opt)
\end{aligned}$$

Hence Proved □

(c) (10 points) In reality the fake theorem is not true as stated above. What is true is the following theorem:

Theorem 2. *Given any metric (X, d) where X is a set of n points, and $d(\cdot)$ is a distance function, we can efficiently find an embedding into a distribution $\mathcal{D}$ over different line metrics $d_{L_1}, d_{L_2}, \ldots, d_{L_k}$ such that for all $u \neq v$, $d(u,v) \leq \mathbb{E}_{d_L \sim \mathcal{D}}[d_L(u,v)] \leq O(\log n) d(u,v)$.*

Now, can you extend the same reasoning to devise good approximation algorithms for k-median? If not, what hurdles do you face? For a hint, see the part below.

Solution:

Proof. We have an expectation in both inequalities, hence we can't actually pick up any random solution and hope that it satisfies the metric inequality of part (b), there is some probability with which it might happen that the value of $d_L(u,v)$ is greater than $O(\log n)d(u,v)$ and there is some probability with which $d_L(u,v)$ might be less than d(u,v), if both these constraints hold true it is difficult to find a set that holds these conditions for all pairs of u,v, it might happen that for some pairs one condition is violated and for some the other condition is.Thus with these constraints a good approximation algorithm is not straightforward, hence these are the hurdles that we face in this part of the problem. □

(d) (10 points) What happens if you further assume in the theorem (in reality this assumption is not valid, but let us just assume it for the sake of this question) that for all line metrics d_{L_i} in the support of $\mathcal{D}$ and all $u \neq v$, we have $d_{L_i}(u,v) \geq d(u,v)$, and $\mathbb{E}_{d_L \sim \mathcal{D}}[d_L(u,v)] \leq O(\log n) d(u,v)$. Such embeddings are called embeddings into a distribution of **dominating lines** (for any pair $u \neq v$, the embedding function *always increases* the distances, but *on expectation*, does not increase by more than a $O(\log n)$ factor). Now does it help you devise approximation algorithms?

Solution:

Proof. Yes now we can devise approximation algorithm as follows, we pick a random line metric and perform our calculations on that metric same as we did in part(a). Here we show that using markov's inequality there is a constant probability that we will find a $\log n$ approximation. From part (a), we find the the first inequality we have the fact that $cost(Alg) \leq cost_{d_{L_i}}(C, Opt_{d_{L_i}})$. To show a $\log n$ approximation we need that $d_L(u,v) \leq O(\log n)d(u,v)$. Using markov's inequality we have,

$$
\begin{aligned}
Pr(X \geq a) &\leq \frac{E[X]}{a} \\
Pr(X \geq 2 * O(\log n) d(u,v)) &\leq \frac{E[X]}{2 * O(\log n) d(u,v)} \\
Pr(d_L(u,v) \geq 2 * O(\log n) d(u,v)) &\leq \frac{E[d_L(u,v)]}{2 * O(\log n) d(u,v)} \\
Pr(d_L(u,v) \geq 2 * O(\log n) d(u,v)) &\leq \frac{1}{2}
\end{aligned}
$$

, From our assumption on the distribution of metrics

Hence we have shown that probability that our randomly chosen metric fails is less than equal to half. Hence if we keep randomly choosing metrics one after another, after a few rounds we would have found a metric that satisfies $d_L(u,v) \leq 2 * O(\log n) d(u,v)$ and $d_L(u,v) \geq d(u,v)$. □

2. (40 points) **(Algorithms for Warehouse Placement)** We now look at a modification of the k-median problem, which we call warehouse placement. Here, we're given a metric space with n retail outlets (or clients) denoted by set C, and m possible locations F, and a metric d over $C \cup F$. We want to place k warehouses at a subset $S \subseteq F$ such that $\max_{j \in C} d(j, S)$ is minimized. Notice that if the max is replaced by $\sum$, then we will get back the k-median problem.

   (a) (10 points) Show that the problem does not admit any $(3 - \epsilon)$-approximation algorithm if $P \neq NP$ (Hint: reduce from Max-k-Coverage).

Solution:

Proof. Proof idea is that we try to reduce the problem in polynomial time from a max k-coverage problem to an instance of the given problem such that if we run a $(3 - \epsilon)$ approximation algorithm on it we would have solved the max-k coverage problem.
We begin the construction, the elements in the max k coverage set are referred to as clients in the warehouse placement problem. The sets that cover these elements are referred to as warehouses/facilities in our problem setting. These warehouses and clients have a distance metric as follows, if a set covers an element then the distance between the corresponding warehouse and client is 1 and if it doesn't cover then the distance between the corresponding warehouse and clients is 3. All the distances between distinct clients is 2 and all the distances between distinct warehouses are also 2. Distances between same clients and same warehouses are zero in both cases. The proof involves two parts, first we show that the distance metric is satisfied and the problem is a valid construc-

tion. Then we show that if we knew a less than 3 approximation algorithm then we could solve the max k coverage problem.

Valid Construction in Metric Space :

$$\begin{aligned} U = \{e_1, e_2, \ldots, e_n\} \Rightarrow C = \{e_1, e_2, \ldots, e_n\} \\ S = \{s_1, s_2, \ldots, s_m\} \Rightarrow F = \{s_1, s_2, \ldots, s_m\} \\ \forall i, d(e_i, e_i) = 0 \\ \forall i, j | i \neq j, d(e_i, e_j) = 2 \\ \forall j, d(s_j, s_j) = 0 \\ \forall i, j | i \neq j, d(s_i, s_j) = 2 \\ \forall u, v, d(u, v) = d(v, u) \\ d(e_i, s_j) = 1 | e_i \in s_j \\ d(e_i, s_j) = 3 | e_i \notin s_j \end{aligned}$$

Symmetric and reflexive are satisfied we need to check if triangle inequality is satisfied. Since we consider the case when all three vertices chosen in the triangle inequality are distinct, if not then all the distances being greater than zero trivially prove the triangle inequality. If all three vertices are distinct the distances possible on the sides of the triangle are $1, 2, 3$, the only case the inequality fails is if there are two sides with length 1 and the third side is of length 3, in all other cases the inequality is satisfied. We argue that this case is not at all possible as if $d(u, v) = 1$ and $d(v, w) = 1$ and $d(u, w) = 3$, then wlog assume u is in C, which implies that w is in F, as distance between same sets is either 0 or 2.And as u is in C, and $d(u, v) = 1$ means v is in F, now v is in F and w is in F but still the distance between them is 1, hence contradiction thus this is a valid metric construction.

Next we observe that if there exists a valid k coverage that is a k collection of sets S such that it covers all the elements then the answer to our problem is 1, we are asked to minimize the max distance between the clients and facilities, this distance is either 3 or 1. If there is a k coverage choose the corresponding facilities as the k warehouses set, as the union covers all the elements, meaning for each client there exists a warehouse with distance 1, hence the optimal is distance 1. Now if there is not a k coverage, that is all possible k sets don't cover the universe, then there exists a element always that is uncovered by our k sets, which means there is always a client that has distance 3 with all the warehouses we have chosen, which means the optimal answer is 3. Now if there existed a less than 3 approximation algorithm then we would run it on our setting and find if the algorithm answer is less than 3 or greater than 3, if the algorithm answer was less than 3, that would mean the optimal answer was 1 and would answer yes to our NP-hard decision problem. And if the algorithm answers greater than 3 that would mean we would answer no to our NP-hard decision problem implying that we have proven P $\neq NP$. □

Now analyze the following algorithm for this problem: suppose somehow we know the value of the optimal solution C^*. Then, build a graph over the clients C with the following edges: place an edge between $j_1 \in C$ and $j_2 \in C$ if and only if $d(j_1, j_2) \leq 2C^*$.

(b) (10 points) Show that the size of any maximal independent set in this graph is at most k.

Solution:

Proof. We will first note the following lemma that if the facility that a client is closest connected to is the same then the distance between those two clients is less than $2C^*$. i.e Let F_1 be a facility, such that for two clients C_1, C_2. $d(C_1, S) = d(C_1, F_1)$ and $d(C_2, S) = d(C_2, F_1)$ where S is the optimal facility subset. Now we prove,

$$d(C_1, S) \leq C^* \Rightarrow d(C_1, F_1) \leq C^*$$
$$d(C_2, S) \leq C^* \Rightarrow d(C_2, F_1) \leq C^*$$

Because C^* is the max over all clients hence its value must be greater than the value at two of the clients.
Using metric properties,

$$\begin{aligned} d(C_2, F_1) &= d(F_1, C_2) \\ d(u, w) &\leq d(u, v) + d(v, w) \\ d(C_1, C_2) &\leq d(C_1, F_1) + d(F_1, C_2) \\ d(C_1, C_2) &\leq 2C^* \end{aligned}$$

Hence C_1 and C_2 cannot be in the same independent set. Hence we have proved that if two clients share the same facility as their closest facility in the optimal solution then they cannot be present in the independent set. If size of independent set is greater than k then by pigeon hole principle there exists two clients that are mapped to the same facility in S. (Because $|S| = k$). Hence there will be two clients forming an edge which is a contradiction to the independent set property. Hence maximum size of any independent set is k. □

(c) (10 points) Show how to use some maximal independent set found to recover a set $S \subseteq F$ of k locations with a good approximation ratio for the original problem. What's the best approximation factor you can get?

Solution:

Proof. We get a 3 approximation for this algorithm. In the maximal independent set, we consider the facility in the set F that each client in the independent set is nearest to. Since we know that the maximal independent set has size at most k, we end up finding at most k facilities from the set F. If the facility is less than k in number then we arbitrarily pick any facility and make the facilities chosen to be k out of the total facilities. We claim that this algorithm is a three approximation algorithm. Here we prove the result. Let S' be the set of facilities picked from the independent set. And S be the final solution of facilities picked after arbitrary addition.

We know that, $Cost(Alg) = max_{j \in C} d(j, S)$. Let there be two sets C_1 indicating the clients that are present in the maximum independent set and C_2 the clients that are not present in the maximum independent set.$Cost(Alg) = max(max_{j \in C_1} d(j, S), max_{j \in C_2} d(j, S))$. As $d(j, S) \leq d(j, S')$, $\Rightarrow$ $Cost(Alg) \leq max(max_{j \in C_1} d(j, S'), max_{j \in C_2} d(j, S'))$.Now for all clients in C_1 we know that the set in F was picked which they were closest to, and this value can't be greater than C^* because if that happened then the closest facility in the whole set is farther than C^* which implies that the value of any feasible solution must be greater than C^* which is a contradiction, hence $\forall_{j \in C_1} d(j, S') \leq C^*$.

Now wlog consider a client in C_2 called u, since its in C_2 it is connected to one of the clients in C_1 otherwise the maximal independent set property is violated, let this client be v. Since v is in C_1 it has the facility that it is nearest to in the set S'. Let this facility be w, we showed above that $d(v, w) \leq C^*$. We know that $d(u, v) \leq 2C^*$. Using triangle inequality,

$$\begin{aligned} d(u, v) + d(v, w) &\geq d(u, w) \\ 3C^* &\geq d(u, w) \end{aligned}$$

This means in our facility set there exists a facility such that distance between u and the facility is less than $3C^*$. This implies that $d(u, S) \leq 3C^*$. As u was chosen without loss of generality, we have showed that for all vertices in C_2 max of them is less than equal to $3C^*$, since vertices in C_1 had a max cost of C^*, hence,

$$cost(Alg) \leq 3C^*$$

Hence a 3 approximation algorithm. □

(d) (10 points) Show how to dispense the assumption of knowing C^* by trying all distance values and enumerating.

Solution:

Proof. Since we are creating a graph based on C^*, the absolute value of C^* doesn't have any significance. If we simply calculate all $\binom{n}{2}$ distances between any two clients and for each pair of distances between clients say u and v we draw the graph G such that an edge exists between vertices j_1 and j_2 if and only if $d(j_1, j_2) \leq d(u, v)$. Then out of the $\binom{n}{2}$ distances one exists such that the graph drawn with $d(j_1, j_2) \leq 2C^*$ is identical. In other words, when we construct the graph with $2C^*$ as my separating distance (Let it be G)then we can increase the separating distance by small values till one of the inequalities get bounded and when this happens the separating distance is same as one of the distances between two vertices. Hence simply picking the minimum of all solutions that we suggested will include the solution with separating distance as $2C^*$ and will generate a $O(n^2)$ blowup in complexity but still be a 3 approximation. □