1. (20 points) **(Weighted Vertex Cover with Penalties)** In the problem we did in class, we are given a graph $G = (V, E)$ with non-negative weights on vertices. The goal is to find a minimum cost vertex cover to cover all edges. Here we consider a relaxed version where each edge has a penalty $\pi_e \geq 0$ if the set of vertices picked does not cover it. The goal is to minimize the total cost of vertices picked plus the total penalty of edges not covered.

   (a) (5 points) Write down a natural LP relaxation for the problem.

   Solution:

   Proof. $1 + 1 = 2$. □

   (b) (5 points) Write down the dual of this LP relaxation.

   Solution:

   Proof. $1 + 1 = 2$. □

   (c) (10 points) Devise a primal-dual 3-approximation for this problem.

   Solution:

   Proof. $1 + 1 = 2$. □

2. (40 points) **(Matching on Metrics)** In the classical perfect matching problem, we are given a graph $G = (V, E)$ on $2n$ vertices with non-negative weights on edges. The goal is to find a minimum cost perfect matching on these $2n$ vertices.

   (a) (3 points) What is the complexity of this problem, i.e., is it in P? If so, what's the best known running time of the algorithm? You can search online to find out more.

   Solution:

   Proof. $1 + 1 = 2$. □

   (b) (3 points) How good is the natural greedy algorithm for this problem? That is, pick the cheapest edge, pair those vertices, and repeat on the remaining edges. How bad is the approximation performance of this algorithm?

   Solution:

   Proof. $1 + 1 = 2$. □

   (c) (10 points) Now suppose the weights satisfy the metric property, i.e., $w(u, v) = w(v, u)$ and $w(u, v) + w(v, w) \geq w(u, w)$ for all u, v, w. Then is the greedy algorithm any good? Can you come up with a bad lower bound? (Hint: think of a recursive construction on a line metric).

Solution:

Proof. $1 + 1 = 2$. □

(d) (4 points) Now we construct a primal-dual algorithm on metric spaces. Consider an LP like the Steiner forest LP we discussed in class, where we want to minimize $\sum_{uv} w(u, v)x_{uv}$ and the constraints are $\sum_{e\in\delta(S)} x_e \geq 1$ for all S of *odd cardinality*. Show that the optimal cost of this LP is at most the optimal matching cost for the problem.

Solution:

Proof. $1 + 1 = 2$. □

(e) (7 points) Write down the dual of the above LP. Then devise a primal-dual algorithm like we did in class with all violated components raising their dual variables, and also with the reverse delete step.

Solution:

Proof. $1 + 1 = 2$. □

(f) (8 points) Prove that the primal-dual framework yields a primal-feasible solution (after reverse delete) with cost at most twice the dual solution's value.

Solution:

Proof. $1 + 1 = 2$. □

(g) (5 points) Are we done? Show that we can convert any primal feasible solution to a perfect matching of no greater cost.

Solution:

Proof. $1 + 1 = 2$. □

3. (20 points) **(Better Local Search for k-Median)** We will get a more detailed understanding of the local search technique for the k-median problem. More precisely, instead of the algorithm discussed in class, imagine if we could do up to t swaps at once, in any local move, and stop when all such neighborhood solutions are non-improving. How much better is this algorithm in its performance?

(a) (3 points) Construct the η graph just like we did for the normal local search, and now partition the vertices in S into the following kinds: (i) $1 \leq |\eta^{-1}(i)| \leq t$, (ii) $|\eta^{-1}(i)| > t$, and (iii) $|\eta^{-1}(i)| = 0$. If there are k_1 centers in S^* which map to the first kind of vertices above, then how large can the set of type (iii) be above in terms of k, k_1 and t?

Solution:

Proof. $1 + 1 = 2$. □

(b) (7 points) Devise a set of valid test swaps so that each center in S^* is used exactly t times, and each center in S is used no more than $t + 1$ times, and moreover, no center of type (ii) is involved in any test swap.

Solution:

Proof. $1 + 1 = 2$. □

(c) (10 points) Formally prove the performance of the algorithm in terms of OPT using these inequalities from the test swaps. Try to get a $3 + O(1/t)$ approximation ratio.

Solution:

Proof. $1 + 1 = 2$. □

4. (0 points) **Difficulty Level.** How difficult was this homework? How much time would you have spent on these questions?

Solution:

Proof. $1 + 1 = 2$. □