Problem Set 1 (due Tuesday, September 26)

- Since the background of the students in the class varies, some may find these problems easy while others may find some problems challenging. The main purpose of these problem sets is for you to learn the course material better by applying ideas learnt in class and/or exploring related problems.
- Please work on these problems on your own, or in collaboration with fellow students in class. Please try not to seek out solutions or solution approaches from the web. If you need hints, post on Piazza; discussions there may be helpful for all.
- Please typeset your solution. Latex, plain text, pdf, and Word are all acceptable formats. For figures, feel free to draw by hand or use any convenient tool.

Problem 1 (1 point) Lower bound instance for the k-center algorithms

Present a family of instances, one for each k, for which Gonzalez's algorithm has cost at least twice the optimal cost.

Present a family of instances, one for each k, for which the Hochbaum-Shmoys algorithm has cost at least twice the optimal cost.

Problem 2 (2 points) Clustering to maximize separation

Many scientific applications, including medical imaging, classification of astronomical objects, and web document categorization, require partitioning a given set of objects into disjoint sets, a problem often referred to as clustering. The clustering problems we are considering in class are NP-complete, in general. Not all objectives are intractable, however. Consider the following variant of clustering. You are given a set S of n points $v_1, \ldots, v_n$, together with their pairwise distances, which form a metric space. So, the distances are symmetric and nonnegative, the distance between a point and itself is zero, while that between two distinct points is nonzero. Let $d(u, v)$ denote the distance between two points u and v. For any two nonempty sets X and Y of points, define the *distance* $d(X, Y)$ *between* X *and* Y to be

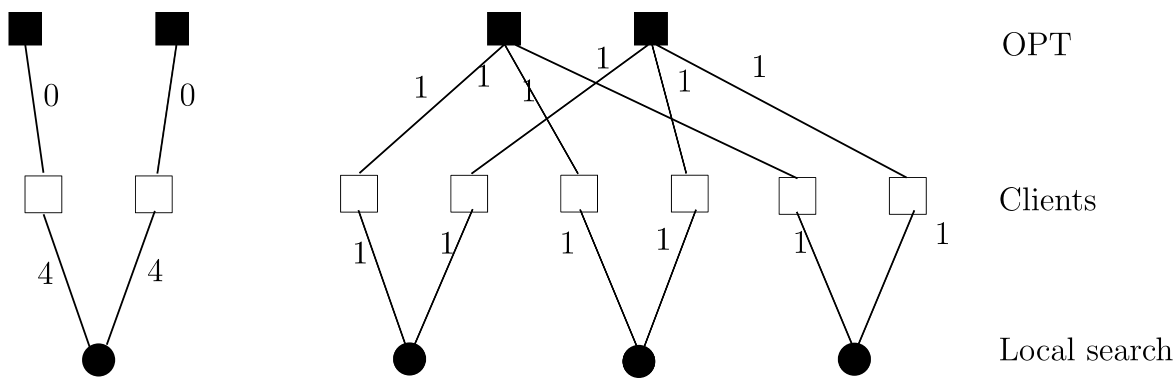
$$\min_{u \in X, v \in Y} d(u, v).$$

Given a number $m \leq n$, you are asked to partition S into m nonempty disjoint sets $S_1, \ldots, S_m$ so as to maximize the minimum distance between any pair of disjoint sets; that is, determine disjoint sets S_1 through S_m that maximizes $\min_{i \neq j} d(S_i, S_j)$ subject to the constraint that $\cup_i S_i = S$.

Give an algorithm to solve the above problem. Analyze the running time. Make your algorithm as efficient as you can, in terms of its worst case running time. (*Hint:* Construct a weighted complete graph and relate the given problem to that of removing edges in this graph to achieve an appropriate objective.)

Problem 3 (1 point) Lower bound instance for the k-median local search algorithm

It is not too difficult to find an example where the local search algorithm for k-median does not yield an optimum solution. It is a bit more challenging to find instances where the approximation ratio of 5 is tight. Consider the following instance. Assume that facilities can only be opened at the shaded locations and the only clients are the clear locations. The metric is given by the shortest path distances in the given graph. (Think of 0 as a really tiny number and disconnected parts having an arbitrarily large distance among them.)



Using the above gadgets, present a family of instances, for infinite values of k, for which a local optimum for k-median, using the local search algorithm studied in class, has cost at least five times the optimal cost.

Problem 4 (1 point) 3-approximation for k-median in a special case

Suppose we find a local optimum S which has the property that there exists a global optimum S^* such that for every $s \in S$, there exists exactly one $s^* \in S^*$ with $\sigma(s^*) = s$. Prove that the cost of S is at most three times the optimal cost.

Problem 5 (2 + 2 + 1 = 5 points) Local search for a uniform coloring problem

Let $G = (V, E)$ be an undirected graph, in which we have to assign a color from a palette C to each vertex. For each color i and each vertex, there is a cost c_v^i to assign color i to v. The aim is to have as many neighboring nodes the same coloring as possible. In particular, for each edge $e = (u, v)$ such that the color of u is different than the color of v, we pay an associated cost c_e. The goal is to minimize the total cost.

In this problem, we consider the following local search algorithm. Given a current coloring, for color i, let an i-*recoloring* be the step in which we find a minimum-cost recoloring where each node either keeps its current color or switches to i. If there exists an i-recoloring that reduces the cost, then the local search algorithm moves to the new coloring; otherwise, we have reached a local optimum.

(a) Show that the i-recoloring step can be solved in polynomial time using an s-t minimum cut algorithm (recall maximum flows and minimum cuts from your algorithms class).

(*Hint:* Using the given graph, build a new network flow graph, with a source s representing color i and sink t representing other colors. Show that a minimum s-t cut in this graph can be associated with a i-recoloring.)

(b) Show that the cost of a local optimal coloring is at most twice that of a global optimum.

(c) Show that for any $\varepsilon > 0$, there is a polynomial-time $(2 + \varepsilon)$-approximation algorithm.