Do not use external sources beyond the materials linked to on the course website to solve these problems.

Remember that you must write up your solutions **independently**, and **list your collaborators** by name clearly at top of your submission (or "no collaborators" if none).

28 points total, with an additional 20 bonus points.

1. (6 pts) Gru is despicably trying to decide which of k different transmitter stations to activate to communicate with his n drones. The cost of activating station j is a_j, and the cost of having drone i communicate with station j is $c_{i,j}$. Given these costs, Gru must find the minimum cost way to ensure that every drone is communicating with at least one station. Formulate this problem as an integer programming problem, and write its linear programming relaxation.

2. (12 total pts) Consider a variant of the classic network flow problem (which you can read about in Chapter 7.2), where there is a (possibly negative) per-unit penalty $p_{u,v}$ associated with every edge (u, v) in addition to the capacity bound $c_{u,v}$, and a target amount of flow $d > 0$ is given. For every unit of flow transmitted along (u, v), a penalty of $p_{u,v}$ is incurred. As usual we have a source node s and a sink node t. The goal is to find an s–t flow of amount at least d that minimizes the total cost, or report that such a flow does not exist.

   a. (6 pts) Show how to solve this problem using linear programming.

   b. (6 pts) Show that the s–t shortest path problem and the classic network flow problem are both special cases of this problem.

3. (10 pts) A graph $G = (V, E)$ is *triangle-free* if it contains no triangles: for every u, v, x at least one of (u, v), (v, x), (u, x) is missing from E. Give a polynomial-time 3-approximation algorithm for the following problem: given an undirected graph G with nonnegative edge weights, find the maximum-weight subgraph which is triangle-free. (In other words, remove a set of edges of minimum total weight from G so the remaining graph is triangle-free.)

4. (**Optional: 10 total bonus pts**) In this problem you wish to minimize a strictly convex function $f : \mathbb{R} \to \mathbb{R}$ given *oracle* access to certain quantities, meaning you can query the oracle for the relevant quantities in $O(1)$ time per call. For example, oracle access to f would mean that each query $f(x)$ takes $O(1)$ time. Give an efficient

algorithm to find an approximate minimizer of f, meaning an x such that $|x - x^*| < \epsilon$ where x^* is the unique minimizer, given oracle access to:

a. (**3 bonus pts**) Both f and its derivative f'.

b. (**3 bonus pts**) Only the derivative f'.

c. (**4 bonus pts**) Only f (do not assume f is differentiable).

5. (**Optional: 10 bonus pts**) You are about to send your trusty biped robot Ada to a mysterious trench in the Kalahari desert to search for an ancient artifact—since the trench is extremely long and narrow, running East-West, this is effectively a one-dimensional search mission. Ada will be placed at the best guess for the location of the artifact, and start searching, and because of the nature of the trench, this search consists only of steps to the East or West. You wish to program Ada with a deterministic algorithm to find the artifact efficiently, with the guarantee that the total number of steps taken is at most a constant times the optimal number of steps. Find such an algorithm and constant, and prove that your constant is the best possible.