Problem Set #10

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **You do not need to turn these problems in. The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Deleting Edges

Consider the following problem. You are given a flow network with unit capacity edges: It consists of a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$; and $c_e = 1$ for all $e \in E$. You are also given a parameter k.

The goal is to delete k edges so as to reduce the maximum s-t flow in G by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum s-t flow in $G' = (V, E - F)$ is as small as possible subject to this.

Give a polynomial time algorithm to solve this problem. Argue (prove) that your algorithm does in fact find the graph with the smallest maximum flow.

Problem 2: Flow Networks and Sensor Failures

An ad hoc sensor network network is made up of low-cost, low-power devices that are distributed in some physical space. One example of these networks in the literature involves tossing a bunch of these sensors out of an airplane over a forest. The sensors are then randomly distributed and expected to monitor for conditions that are indicative of a forest fire.

The sensor nodes, however, are prone to failure for a variety of reasons. A given sensor s can detect that it is about to fail (in practice, this may not be possible, but let's assume it's true). When s determines it is failing, it needs to send a representation of its current state to some other nearby sensor that can take over for it. Each device has a limited transmission range, d; i.e., a node can communicate with exactly the set of nodes within d meters. (Notice that this relationship is symmetric: if sensor s_1 can communicate with s_2, the reverse is also true.) Because some set of the sensor's neighbors may have already failed, we actually want to make sure that we send the backup copy to some live node; therefore when a sensor s is failing, it should send its state to k other sensors. These k sensors constitute the *back up set* for s.

You're given a set of n sensors with known positions represented by an (x, y) coordinate for each sensor. You must design an algorithm that determines whether it is possible to choose a back up set for each of the n devices (i.e., for each device, find k other sensors with d meters), with the added constraint that a single sensor can serve as a back up for at most b other sensors. You must also output the back up sets for each sensor, given that a solution exists.

To be completely clear, ensure you complete the following steps:

- Formulate the problem as a network flow problem. This means define the vertices, edges, and capacities of the network flow graph *and* relate your rationale for the structure of the network flow graph to the original problem.
- Be sure to represent k and b in your network flow graph.
- Given a max-flow or a min-cut on your network flow graph, describe how you can determine whether it is possible to choose a back up set for each sensor.
- Given a max-flow or a min-cut on your network flow graph, describe how to state the back up sets, given that a feasible solution exists.

Problem 3: Zero-Weight Cycle

You are give a directed graph $G = (V, E)$ with weights w_e on its edges $e \in E$. The weights can be negative or positive. The *Zero-Weight Cycle* Problem is to decide if there is a simple cycle in G so that that sum of the edge weights on this cycle is exactly 0. Prove that *Zero-Weight Cycle* is NP-Complete by reducing from the subset sum problem.

The Subset Sum Problem. *Given natural numbers $w_1, w_2, \ldots, w_n$ and a target number W, is there a subset of $\{w_1, w_2, \ldots, w_n\}$ that adds up to precisely W?*

Problem 4: Efficient Recruiting

Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who is skilled at each of the n sports covered by the camp (baseball, volleyball, etc.). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the *Efficient Recruiting* Problem. Show that *Efficient Recruiting* is NP-Complete.

The Vertex Cover Problem. *Given a graph G and a number k, does G contain a vertex cover of size at most k? (Recall that a vertex cover $V' \subseteq V$ is a set of vertices such that every edge $e \in E$ has at least one of its endpoints in V'.)*

Problem 5: Integer Knapsack Problem

Prove that the integer knapsack problem is NP-complete, by a reduction from the subset-sum problem, defined in Problem 3.