TTIC 31010 / CMSC 37000 Algorithms, Winter Quarter 2019

Homework # 4 **Due: February 28, 2019**

Homeworks due in class on the due date. Please put your name on each page of your handin just in case pages get separated.

Lateness policy: up to 24 hours late: 10 points off. 24-48 hours late: 20 points off. More than 48 hours late: 60 points off (at this point, solutions will be posted - and you may look at them if you wish, but your answers should be in your own words).

Written homeworks are to be done *individually*. Group work is only for the oral-presentation assignments. If you have questions, please contact the course staff.

Please do *not* post questions or solutions on websites such as coursehero etc.

Problems:

(30 pts) 1. **Multicommodity Flow.** The *multicommodity flow* problem is just like the standard network flow problem except we have p sources $s_1, \ldots, s_p$ and p sinks $t_1, \ldots, t_p$. The materials flowing from s_1 have to go to t_1, the materials from s_2 have to go to t_2, and so on. For each sink t_i we have a *demand* d_i. (That is, we need to get d_1 units of commodity 1 from s_1 to t_1, d_2 units of commodity 2 from s_2 to t_2, and so on.) Our goal is to solve for a *feasible solution* — a solution satisfying the demands — if one exists. (Just like with standard network flow, the total amount of materials going on some edge (u, v) cannot exceed its capacity c_{uv}. However, our "flow-in = flow-out" constraints must hold separately for each commodity. That is, for every commodity i, and every vertex $v \notin \{s_i, t_i\}$, the amount of type-i materials going into v must equal the amount of type-i materials going out from v.

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

(b) The above problem assumes all edges are directed. E.g., if you had a highway with 3 lanes going one way and two lanes going the other, that would be a directed edge of capacity 3 in one direction and a directed edge of capacity 2 in the other. Suppose we wanted to also allow undirected edges e with capacities c_e (like a highway with 5 lanes where part of your job is to decide how many will go one way and how many will go the other). How can you modify your LP formulation to handle this as well?

(35 pts) 2. **Graduation Revisited.** The University of Bureaucracy has has switched to a less draconian policy for graduation requirements than that used on the last homework. As in Homework 3, there is a list of requirements where requirement i is of the form: "you must take at least k_i courses from set S_i". However, unlike the case in Homework 3, a student *may* use the same course to fulfill several requirements. For example, if one requirement stated that a student must take at least one course from $\{A, B, C\}$, another required at least one course from $\{C, D, E\}$, and a third required at least one course from $\{A, F, G\}$, then a student would only have to take A and C to graduate.

This change now makes the problem from the previous homework much easier: given a list L of courses taken by some student, it is now very easy to tell if they can graduate. But, now we want to answer a different question. Specifically, consider an incoming freshman interested in finding the *minimum* number of courses that they need to take in order to graduate.

(a) Prove that the problem faced by this freshman is NP-hard, even if each k_i is equal to 1. Specifically, consider the following decision problem: given n items labeled $1, 2, \ldots, n$, given m subsets of these items $S_1, S_2, \ldots, S_m$, and given an integer k, does there exist a set S of at most k items such that $|S \bigcap S_i| \geq 1$ for all S_i. Prove that this problem is NP-complete (also say why it is in NP).

(b) Show how you could use a polynomial-time algorithm for the above decision problem to also solve the search-version of the problem (i.e., actually find a minimum-sized set of courses to take).

(c) We could define a *fractional* version of the graduation problem by imagining that in each course taken, a student can elect to do a fraction of the work between 0.00 and 1.00, and that requirement i now states "the sum of your fractions of work in courses taken from set S_i must be at least k_i" (courses not taken count as 0). The student now wants to know the least total work needed to satisfy all requirements and graduate.

Show how this problem can be solved using *linear programming*. Be sure to specify what the variables are, what the constraints are, and what you are trying to minimize or maximize.

(35 pts) 3. **Splitting Sets.** The *Set-Splitting* problem is the following: given n items $1, \ldots, n$ and m subsets $S_1, \ldots, S_m$ of these items, is it possible to color the items red and blue so that each set S_i contains at least one red item and at least one blue item?

(a) Explain why this problem is in NP.

(b) Prove this problem is NP-hard by a reduction from 3SAT. In particular, we want a way to convert an instance of 3SAT into an instance of set-splitting such that a YES instance of 3SAT gets mapped to a YES instance of set-splitting and a NO instance of 3SAT gets mapped to a NO instance of set-splitting.

Hint: Given a 3SAT formula f on n variables, create $2n + 1$ items: one to "represent" each x_i, one to "represent" each $\bar{x}_i$, and one extra item.