600.363 Introduction to Algorithms / 600.463 Algorithms I **Fall 2014**

Homework #8 **Due:** October November 11, 2014, 1:30pm

Please start each problem on a new page, and include your name on each problem. You can submit on blackboard, under student assessment.

Remember: you may work in groups of up to three people, but must write up your solution entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. You may use the internet to look up formulas, definitions, etc., but may not simply look up the answers online.

Please include proofs with all of your answers, unless stated otherwise.

1 Graduation Requirements (33 points)

John Hopskins University[1] has n courses. In order to graduate, a student must satisfy several requirements of the form "you must take at least k courses from subset S". However, any given course cannot be used towards satisfying multiple requirements. For example, if one requirement says that you must take at least two courses from $\{A, B, C\}$, and a second requirement states that you must take at least two courses from $\{C, D, E\}$, then a student who has taken just $\{B, C, D\}$ would not yet be able to graduate as C can only be used towards one of the requirements.

Your job is to give a polynomial-time algorithm for the following problem: given a list of requirements $r_1, r_2, \ldots, r_m$ (where each requirement r_i is of the form "you must take at least k_i courses from set S_i"), and given a list L of courses taken by some student, determine if that student can graduate.

Hint: think flow.

2 Realizable Degree Sequences (33 points)

(a) Suppose you are given the following set of degree constraints for a four-node graph:

$$\begin{aligned} d_{1,in} &= 0, d_{1,out} = 2 \\ d_{2,in} &= 1, d_{2,out} = 2 \\ d_{3,in} &= 1, d_{3,out} = 1 \\ d_{4,in} &= 3, d_{4,out} = 0 \end{aligned}$$

Is there a directed graph, with no multi-edges or self-loops, that satisfies these constraints? If so, what is it? If not, why not?

(b) This type of specification, where we are given the in- and out-degrees of every vertex, is called a *degree sequence*. The question above was whether a particular degree sequence is *realizable* – that is, whether there exists a graph having those degrees.

Give a polynomial-time algorithm that, given a degree sequence, will determine if it is realizable and if it is will produce a directed graph with those degrees. The graph should not have

[1] `https://www.youtube.com/watch?v=JEH2ha1p0WA`

any self loops or multi-edges (i.e. for each directed pair (i, j) with $i \neq j$ there can be at most one edge from i to j, although it is fine if there is also an edge from j to i).

Hint: think flow.

3 Circulations (33 points)

Let's look at some variations of max-flow. In class, we talked about algorithms to compute the maximum flow from a source s to a sink t in a directed graph $G = (V, E)$ when we are given edge capacities $c : E \to \mathbb{R}^+$. A valid flow was a function $f : E \to \mathbb{R}_{\geq 0}$ so that $f(e) \leq c(e)$ for all $e \in E$, and for all nodes $v \in V \setminus \{s, t\}$ we required the flow in to equal the flow out, i.e. $\sum_u f(u, v) = \sum_u f(v, u)$. (In class we modified this a bit to add skew-symmetry, but that was a mathematical convenience – this is the original and equivalent definition).

(a) What if we no longer have a single s and a single t? Instead, think of every node v as having a demand d_v. If d_v is positive then v wants flow; it is a sink. If d_v is negative then v wants to get rid of flow; it is a source. And if $d_v = 0$ then v wants to simply transit flow; it is neither a source nor a sink. A valid *circulation* is an assignment $f : E \to \mathbb{R}_{\geq 0}$ such that $f(e) \leq c(e)$ for all $e \in E$ (i.e. capacities are not violated), and $\sum_u f(u, v) - \sum_u f(v, u) = d_v$ for all $v \in V$ (i.e. at every node the flow-in minus the flow-out is equal to the demand).

Given a directed graph with capacities and demands, give a polynomial-time algorithm to determine whether a valid circulation exists. Hint: reduce to max-flow.

(b) Let's make things even more complicated. Capacities are an upper bound on flow – we are not allowed to send more flow through an edge than its capacity allows. What if we also want to have lower bounds? That is, in addition to capacities c, we are also given lower bounds $\ell : E \to \mathbb{R}_{\geq 0}$ and we require $\ell(e) \leq f(e) \leq c(e)$ for all $e \in E$?

Let's redefine a valid circulation with these lower bounds. We are now given a directed graph $G = (V, E)$, capacities $c : E \to \mathbb{R}^+$, lower bounds $\ell : E \to \mathbb{E}_{\geq 0}$, and for each vertex v we are given a demand d_v. A valid circulation is now a function $f : E \to \mathbb{R}_{\geq 0}$ such that $\ell(e) \leq f(e) \leq c(e)$ for all $e \in E$, and $\sum_u f(u, v) - \sum_u f(v, u) = d_v$.

Give a polynomial-time algorithm to determine whether a valid circulation exists.

Hint: Try to transform the problem into find a circulation without lower bounds, i.e. the setting of part (a). You might want to start by initially setting $f(e) = \ell(e)$ for all e. This will not be a valid circulation, but what does the "remaining" problem look like?