University of Washington
Department of Computer Science and Engineering
CSE 417, Winter 2020

March 3, 2020

Homework 9, Due Friday, March 13, 2020

Problem 1 (10 points):

(Page 324, Exercise 13.) The problem of searching for cycles in graphs arises naturally in financial trading applications. Consider a firm that trades shares in n different companies. For each pair $i \neq j$, they maintain a trade ratio r_{ij}, meaning that one share of i trades for r_{ij} shares of j. Here we allow the rate r to be fractional; that is, $r_{ij} = \frac{2}{3}$ means that you can trade three shares of i to get two shares of j.

A *trading cycle* for a sequence of shares $i_1, i_2, \ldots, i_k$ consists of successively trading shares in company i_1 for shares in company i_2, then shares in company i_2 for shares i_3, and so on, finally trading shares in i_k back to shares in company i_1. After such a sequence of trades, one ends up with shares in the same company i_1 that one starts with. Trading around a cycle is usually a bad idea, as you tend to end up with fewer shares than you started with. But occasionally, for short periods of time, there are opportunities to increase shares. We will call such a cycle an *opportunity* cycle, if trading along the cycle increases the number of shares. This happens exactly if the product of the ratios along the cycle is above 1. In analyzing the state of the market, a firm engaged in trading would like to know if there are any opportunity cycles.

Give a polynomial-time algorithm that finds such an opportunity cycle, if one exists.

Problem 2 (10 points):

Give an algorithm, which given a directed graph $G = (V, E)$, with vertices $s, t \in V$ and an integer k, determines the number of paths from s to t of length k. Your algorithm should be polynomial in k, $|V|$ and $|E|$.

Problem 3 (10 points):

Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

Let G be an arbitrary flow network, with a source s, a sink t and a positive integer capacity c_e on every edge e; and let (A, B) be a minimum $s-t$ cut with respect to these capacities $\{c_e : e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum $s-t$ cut with respect to these new capacities $\{1 + c_e : e \in E\}$.

Problem 4 (10 points):

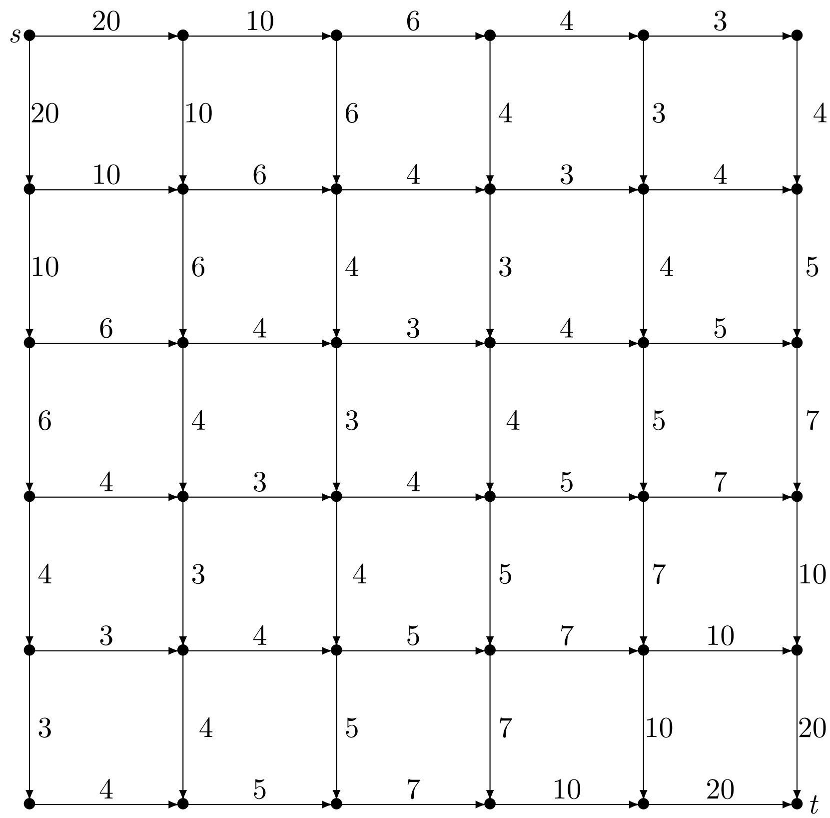
In a standard $s-t$ Maximum-Flow Problem, we assume edges have capacities, and there is no limit on how much flow is allowed to pass through a node. In this problem, we consider the variant of the Maximum-Flow problem with node capacities.

Let $G = (V, E)$ be a directed graph, with source $s \in V$, sink $t \in V$, and nonnegative node capacities $\{c_v \geq 0\}$ for each $v \in V$. Given a flow f in this graph, the flow through a node v is defined as $f^{\text{in}}(v)$, the sum of the flows on the incoming edges to v. We say that a flow is feasible if it satisfies the usual flow-conservation constraints and the node-capacity constraints: $f^{\text{in}}(v) \leq c_v$ for all nodes.

Give a polynomial-time algorithm to find an $s-t$ maximum flow in such a node-capacitated network. Justify the correctness of your algorithm.

Problem 5 (10 points):

Consider the following flow graph. Find a maximum flow.



a) What is the value of the maximum flow? Indicate the value of flow on each edge.

b) Prove that your flow is maximum.

Problem 6 (10 Points):

(Kleinberg-Tardos, Based on exercise 9, Page 419) Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.

Consider the following scenario. Due to a virus outbreak in a region, paramedics have identified a set of n infected people distributed across the region who need to be rushed to hospitals. There are

k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

At the same time, one does not want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the sick people in such a way that the load on the hospitals is balanced: Each hospital receives at most $\lceil n/k \rceil$ people.

Give a polynomial-time algorithm that takes the given information about the people's locations and determines whether this is possible.