University of Washington March 6, 2019
Department of Computer Science and Engineering
CSE 421, Winter 2019

Homework 8, Due Wednesday, March 15, 2019

Problem 1 (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 2 (10 points):

Give an $O(n^2)$ algorithm for finding the longest strictly increasing subsequence of a sequence of n integers. (Note that this problem can be solved in $O(n \log n)$ time by a non-dynamic programming style algorithm, but you do not need find it. Use of dynamic programming for this problem is recommended, but not required.)

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):

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 every $e \in E$. 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, and justify that your algorithm is correct.

Problem 5 (10 points):

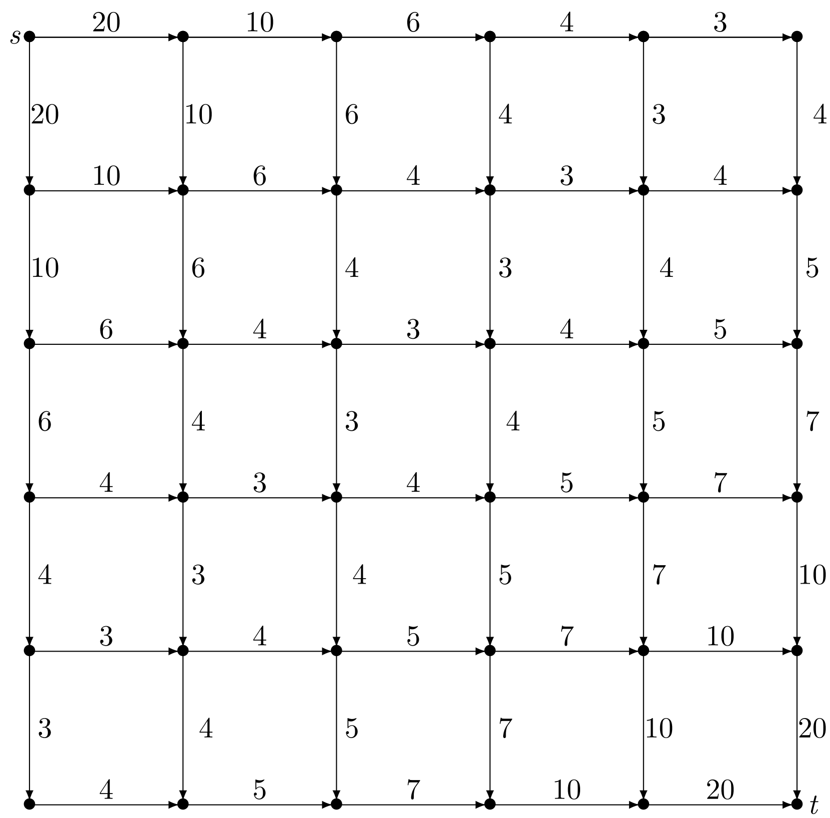
In a standard $s - t$ Maximum-Flow Problem, we assume edges have capacities, and there is no limit on how much flow is allowed to pass through a node. In this problem, we consider the variant of the Maximum-Flow problem with node capacities.

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 $f^{\text{in}}(v)$. 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 6 (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 7 (10 points):

Suppose that you are a wealthy individual who wishes to make campaign donations to candidates $C_1, \ldots, C_n$. You want to donate b_i dollars to candidate C_i. However, campaign finance laws prohibit you from donating to these candidates, so you decide that you will have your friends, $F_1, \ldots, F_m$ help you out by laundering the donations. You determine, for each friend F_i and candidate C_j, the maximum amount a_{ij} that F_i is willing to contribute to C_j for you. Since your friends are not fully trustworthy, you decide to cap the amount of money that you will give to each of them, so you set a bound of M_i for the amount of money you will give to F_i to distribute.

Give a network flow based algorithm which determines how the money is distributed. The algorithm should indicate how much money each friend gives to each candidate. Justify the correctness of your algorithm.