CS515: Algorithms and Data Structures, Fall 2014

Homework 3[*]

Due: Wed, Nov/26/14

Homework Policy: Students should work on homework assignments in group of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in the class. You are allowed to discuss the homework with other groups, however, you must mention their names in your submission. Also, you must cite any other source that you use.

The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.

Problem 1. Suppose you have already computed a maximum flow f^* in a flow network G with integer edge capacities.

(a) Describe and analyze an algorithm to update the maximum flow after the capacity of a single edge is increased by 1.

(b) Describe and analyze an algorithm to update the maximum flow after the capacity of a single edge is decreased by 1.

Both algorithms should be significantly faster than recomputing the maximum flow from scratch.

Problem 2. Let (S, T) and (S', T') be minimum (s, t)-cuts in some flow network G. Prove that $(S \cap S', T \cup T')$ and $(S \cup S', T \cap T')$ are also minimum (s, t)-cuts in G.

Problem 3. *Ad-hoc* networks are made up of low-powered wireless devices. In principle, these networks can be used on battlefields, in regions that have recently suffered from natural disasters, and in other hard-to-reach areas. The idea is that a large collection of cheap, simple devices could be distributed through the area of interest (for example, by dropping them from an airplane); the devices would then automatically configure themselves into a functioning wireless network.

These devices can communicate only within a limited range. We assume all the devices are identical; there is a distance D such that two devices can communicate if and only if the distance between them is at most D.

[*] Problems are from Jeff Erickson's lecture notes. Looking into similar problems from his lecture notes on maximum flow, its applications and linear programming is recommended.

We would like our ad-hoc network to be reliable, but because the devices are cheap and low-powered, they frequently fail. If a device detects that it is likely to fail, it should transmit its information to some other backup device within its communication range. We require each device x to have k potential backup devices, all within distance D of x; we call these k devices the backup set of x. Also, we do not want any device to be in the backup set of too many other devices; otherwise, a single failure might affect a large fraction of the network.

So suppose we are given the communication radius D, parameters b and k, and an array $d[1..n, 1..n]$ of distances, where $d[i, j]$ is the distance between device i and device j. Describe an algorithm that either computes a backup set of size k for each of the n devices, such that no device appears in more than b backup sets, or reports (correctly) that no good collection of backup sets exists.

Problem 4.

(a) Give a linear-programming formulation of the maximum-cardinality bipartite matching problem. The input is a bipartite graph $G = (U \cup V, E)$, where $E \subseteq U \times V$; the output is the largest matching in G. Your linear program should have one variable for each edge.

(b) Now dualize the linear program from part (a). What do the dual variables represent? What does the objective function represent? What problem is this!?