CS515: Algorithms and Data Structures
Final Exam

The exam is TWO pages.

Problem 1. Let $G = (V, E)$ be a directed graph, and let $s, t \in V$ be two vertices of G. For every $(u \to v) \in E$, let $\ell_{u \to v}$ denote the length of the edge $(u \to v)$. Recall the following LP formulation for the (s, t)-shortest path problem.

$$\begin{array}{lll} \text{maximize} & d_t - d_s & \\ \text{subject to} & d_v - d_u \leq \ell_{u \to v} & \text{for every} \quad (u \to v) \in E \\ & d_v \geq 0 & \text{for every} \quad v \in V \end{array}$$

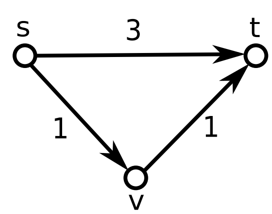
(a) What is the number of variables and the number of constraints of this LP?

(b) Write the dual of this LP? What is the number of variables and constraints of the dual?

We can reduce the number of variables of this LP by setting $d_s = 0$ as follows:

$$\begin{array}{lll} \text{maximize} & d_t & \\ \text{subject to} & d_v - d_u \leq \ell_{u \to v} & \text{for every} \quad (u \to v) \in E \\ & d_v \geq 0 & \text{for every} \quad v \in V \\ & d_s = 0 & \end{array}$$

Consider this new LP for the following graph. ($\ell_{(s \to t)} = 3$, $\ell_{(s \to v)} = 1$, $\ell_{(v \to t)} = 1$.)



(c) Show all constraints and the feasible region by drawing a figure.

(d) Show the objective function direction and the optimum solution in your figure.

(e) Show that $(d_t, d_v) = (0, 0)$ is a feasible point, and show a possible sequence of feasible points that the simplex algorithm can possibly take to reach the optimum from $(d_t, d_v) = (0, 0)$.

(f) Show that $(d_t, d_v) = (3, 0)$ is a locally optimal point, and show a possible sequence of locally optimal points that the simplex algorithm can possibly take to reach the optimum from $(d_t, d_v) = (3, 0)$.

Problem 2. Let $G = (V, E)$ be a directed graph with edge capacities $c : E \to \mathbb{R}_{\geq 0}$, a source vertex s, and a target vertex t. Suppose someone hands you an arbitrary function $f : E \to \mathbb{R}_{\geq 0}$. Describe and analyze fast and simple algorithms to answer the following questions:

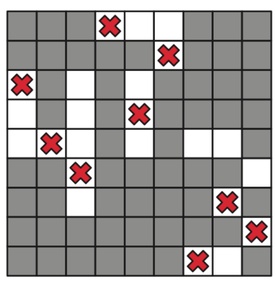
(a) Is f a feasible (s, t)-flow in G?

(b) Is f a maximum (s, t)-flow in G? (You algorithm must be faster than computing maximum flow.)

(c) Is f the **unique** maximum (s, t)-flow in G?

Problem 3. Suppose we are given an $n \times n$ square grid, some of whose squares are colored black and the rest white. Describe and analyze a polynomial time algorithm to determine whether tokens can be placed on the grid so that

1. every token is on a white square;
2. every row of the grid contains exactly one token; and
3. every column of the grid contains exactly one token.



Your input is a two dimensional array IsWhite$[1..n, 1..n]$ of booleans, indicating which squares are white. (IsWhait$[i, j]$ is TRUE if and only if the square at row i and column j is white.) Your output is a single boolean. For example, given the grid above as input, your algorithm should return TRUE.

Problem 4. [Extra Credit] Let $G = (V, E)$ be a *directed graph* and let $s \in V$ be a vertex of G. For each edge $(u \to v) \in E$, let $c_{(u \to v)}$ be a nonnegative cost of destroying $(u \to v)$. If an attacker destroys a set $A \subseteq E$ of edges, he/she receives a nonnegative benefit b_v for each vertex $v \in V$ that can no longer be reached from s. (Of course, benefits of different vertices can be different.) The attacker wants to choose A to minimize cost minus benefit. Suggest an algorithm.