Final Exam, CS 515

Problem 1. Consider the following linear program.

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 3 \\ & x_1 - x_2 \leq 1 \\ & x_1, x_2 \geq 1 \end{aligned}$$

(a) [4 pts] Show all constraints and the feasible region by drawing a figure.

(b) [4 pts] Show the direction of the objective function.

(c) [4 pts] List all feasible vertices and all locally optimal vertices.

(d) [8 pts] Consider $(x_1, x_2) = (0, -1)$ as a vertex. Note that it is not feasible, and it is not locally optimal. Explain, how we can find the optimal solution of the LP starting from this vertex and using both versions of the simplex. Specify, a sequence of points that the simplex might iterate through to achieve the optimal solution from $(0, -1)$.

(e) [10 pts] Write the dual of this LP.

(f) [2 pts] Show all constraints and the feasible region of the dual by drawing a figure (in a separate figure from part (a)).

(g) [2 pts] Show the direction of the objective function in the dual.

(h) [2 pts] List all feasible vertices and all locally optimal vertices in the dual.

(i) [4 pts] Each vertex of the primal corresponds to a vertex of the dual. In particular, the sequence of vertices of part (d) corresponds to a sequence of vertices in the dual LP. Specify this corresponding sequence of vertices in the dual.

Problem 2. Suppose you are given a directed graph $G = (V, E)$ with capacities $c : E \to \mathbb{Z}^+$ and a maximum flow $f : E \to \mathbb{Z}^+$ from some vertex s to some other vertex t in G.

(a) [10 pts] Describe an efficient algorithm to build the residual graph ($G_f = (V, E_f)$) and the residual function ($c_f : E \to \mathbb{R}_{\geq 0}$).

Describe and analyze efficient algorithms for the following operations:

(b) [10 pts] Increment(e): Increase the capacity of edge e by 1 and update the maximum flow f.

(c) [15 pts] Decrement(e): Decrease the capacity of edge e by 1 and update the maximum flow f.

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

Problem 3. [25 pts] An $n \times n$ grid is an undirected graph with n^2 vertices organized into n rows and n columns. We denote the vertex in the ith row and the jth column by (i, j). Every vertex in the grid have exactly four neighbors, except for the boundary vertices, which are the vertices (i, j) such that $i = 1$, $i = n$, $j = 1$, or $j = n$.

Let $x_1, x_2, \ldots, x_k$ be distinct vertices, called terminals, in the $n \times n$ grid. The quick escape problem is to find k vertex-disjoint paths in the grid that connect the terminals to any k distinct boundary vertices. Describe and analyze an efficient algorithm to solve the escape problem. Prove that your algorithm is correct.