Note: Your TA probably will not cover all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. They are deliberately made long so they can serve as a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 LP Basics

Linear Program. A linear program is an optimization problem that seeks the optimal assignment for a linear objective over linear constraints. Let \( x \in \mathbb{R}^n \) be the set of variables and \( A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n \). The canonical form of a linear program is

\[
\text{minimize } c^\top x \\
\text{subject to } Ax \geq b \\
x \geq 0
\]

Any linear program can be written in canonical form.

Let’s check this is the case:

(i) What if the objective is maximization?
(ii) What if you have a constraint \( Ax \leq b \)?
(iii) What about \( Ax = b \)?
(iv) What if the constraint is \( x \leq 0 \)?
(v) What about unconstrained variables \( x \in \mathbb{R} \)?

Dual. The dual of the canonical LP is

\[
\text{maximize } b^\top y \\
\text{subject to } A^\top y \leq c \\
y \geq 0
\]

Weak duality: The objective value of any feasible dual \( \leq \) objective value of any feasible primal

Strong duality: The optimal objective values of these two are equal.

Both are solvable in polynomial time by the Ellipsoid or Interior Point Method.

2 Huffman and LP

Consider the following Huffman code for characters \( a, b, c, d \): \( a = 0, b = 10, c = 110, d = 111 \).

Let \( f_a, f_b, f_c, f_d \) denote the fraction of characters in a file (only containing these characters) that are \( a, b, c, d \) respectively. Write a linear program with variables \( f_a, f_b, f_c, f_d \) to solve the following problem: What values of \( f_a, f_b, f_c, f_d \) that can generate this Huffman code result in the Huffman code using the most bits per character?
3 Job Assignment

There are $I$ people available to work $J$ jobs. The value of person $i$ working 1 day at job $j$ is $a_{ij}$ for $i = 1, \ldots, I$ and $j = 1, \ldots, J$. Each job is completed after the sum of the time of all workers spend on it add up to be 1 day, though partial completion still has value (i.e. person $i$ working $c$ portion of a day on job $j$ is worth $a_{ij}c$). The problem is to find an optimal assignment of jobs for each person for one day such that the total value created by everyone working is optimized. No additional value comes from working on a job after it has been completed.

(a) What variables should we optimize over? I.e. in the canonical linear programming definition, what is $x$?

(b) What are the constraints we need to consider? Hint: there are three major types.

(c) What is the maximization function we are seeking?

4 String Shuffling

Let $x$, $y$, and $z$ be strings. We want to know if $z$ can be obtained only from $x$ and $y$ by interleaving the characters from $x$ and $y$ such that the characters in $x$ appear in order and the characters in $y$ appear in order. For example, if $x = \text{efficient}$ and $y = \text{ALGORITHM}$, then it is true for $z = \text{effALGiORcienTHMt}$, but false for $z = \text{efficientALGORITHMS}$ (extra characters), $z = \text{effALGORITHMMicien}$ (missing the final t), and $z = \text{effOALGRicieITHMnt}$ (out of order). How can we answer this query efficiently? Your answer must be able to efficiently deal with strings with lots of overlap, such as $x = aaaaaaaaaab$ and $y = aaaaaaaac$.

1. Design an efficient algorithm to solve the above problem and state its runtime.

2. Consider an iterative implementation of our DP algorithm in part (a). Naively if we want to keep track of every solved sub-problem, this requires $O(|x||y|)$ space (double check to see if you understand why this is the case). How can we reduce the amount of space our algorithm uses?