Homework # 2

Hard copy is due in class on Wednesday, Nov 8, 2017.

The same guidelines as for Homework 1 apply here. Two questions will be chosen for grading.

1. Let $G$ be a directed graph in which a non-negative weight $w(e)$ is associated with each edge $e$. The length of a path is the sum of its weights. A path of minimum length between two nodes $s$ and $t$ of $G$ is a shortest path.

(a) Explain why the following linear program computes the length of the shortest path from $s$ to $t$.

There is one variable $l_v$ for each node $v$ of $G$.

\[
\begin{align*}
\text{maximize} & \quad l_t \\
\text{subject to} & \quad l_s = 0, \\
& \quad l_v - l_u \leq w(u, v), \text{ for every edge } (u, v) \text{ of } G.
\end{align*}
\]

(b) Given the values $l_v$ for all $v$, how can the shortest path be reconstructed?

2. (From Will Evans) Players Row and Col each choose an integer $x$ and $y$ between 1 and $n$ inclusive. If $x < y - 1$ or $x = y + 1$, player Row wins. If $x > y + 1$ or $x = y - 1$, player Col wins. If $x = y$, no one wins. What is a smallest optimal strategy for player Row and why is it optimal for $n = 2$? For $n = 3$? For $n > 3$?

A smallest optimal strategy is an optimal strategy that places positive probability on the fewest number of choices.

3. (From Dan Gusfield) Suppose that you must dynamically maintain a suffix tree for a string that is being extended over time. You initially build the suffix tree for an initial string $S$. Describe how you would update the suffix tree in each of the following scenarios, which may be repeated often: (a) the string is extended on the left end by one character, (b) the string is extended on the right end by one character.

4. (From Carl Kingsford) Design an algorithm to find the longest repeated substring of a string $S$ such that at least two copies of the substring do not overlap in $S$. 