George Washington University
Department of Computer Science

Csci 6212 - Homework 7

Given: March 30, 2017
Due: 6pm, April 6, 2017

1. Let G be a directed weighted graph, where $V(G) = \{v_1, v_2, \cdots, v_n\}$. Let B be an $n \times n$ matrix such that entry b_{ij} denotes the distance in G from v_i to v_j (using a directed path). Now we are going to insert a new vertex v_{n+1} into G. Let w_i denote the weight of the edge (v_i, v_{n+1}) and w'_i denote the weight of the edge (v_{n+1}, v_i). (If there is no edge from v_i to v_{n+1} or from v_{n+1} to v_i, then w_i or w'_i is inf, respectively.) Describe an algorithm to construct an $(n+1) \times (n+1)$ distance matrix B' from B and values of w_i and w'_i for $1 \leq i \leq n$. (Note that the graph G itself is not given.) Your algorithm should work in $O(n^2)$ time. (Hint: Use the Floyd's dynamic programming algorithm for finding all pairs shortest paths.)

2. Given two strings X and Y, respectively, of length m and n defined over a set $\Sigma = \{a_1, a_2, \cdots, a_k\}$ of finitely many symbols, we are interested in computing an *optimal* (i.e., minimum cost) alignment of two strings, where two possible alignments are defined as (i) a mismatch with cost c_m and (ii) a gap with cost d_g.

   Consider the following two sequences defined over $\Sigma = \{A, G, C, T\}$, where

   $X = \{A\ A\ C\ A\ G\ T\ T\ A\ C\ C\}$ and

   $Y = \{T\ A\ A\ G\ G\ T\ C\ A\}$

   In the following alignment, there are two mismatches and four gaps with total cost $2c_m + 4c_g$:

   $\{-\ A\ A\ C\ A\ G\ T\ T\ A\ C\ C\}$ and

   $\{T\ A\ A\ -\ G\ G\ T\ -\ -\ C\ A\}$.

   Give an dynamic programming algorithm to solve this problem.

3. You are given a boolean expression consisting of a string of the symbols 'true', 'false', 'and', 'or', and 'xor'. Count the number of ways to parenthesize the expression such that it will evaluate to true. For example, there is only 1 way to parenthesize 'true and false xor true' such that it evaluates to true. Give a dynamic programming algorithm to solve this problem and analyze the time complexity of your algorithm.

4. The input to this problem is a sequence S of n integers (not necessarily positive). The problem is to find the consecutive subsequence of S with maximum sum. Consecutive means that you are not allowed to skip numbers. For example if the input was $12, -14, 1, 23, -6, 22, -34, 13$, the output would be $1, 23, -6, 22$. Give a linear time algorithm for this problem.