6.890 HOMEWORK 2 **Due: October 12, 11:59pm**
Submit by email to `6890mit@gmail.com` Handed out: September 28, 2021

Instructions: Everyone needs to submit their own write-up. If you work together with other students, indicate their names on your write-up.

The possible grades for each (sub)problem are Check-, Check and Check+, corresponding to 0, 1 and 2 points respectively. For this problem set the maximum number of points is 20.

Problem 0

Give an $\tilde{O}(n^2)$ time algorithm for the following problem. Given a directed graph $G = (V, E)$ on n nodes, with integer edge weights and a vertex $s \in V$, compute for all $u, v \in V$, the minimum last edge weight of a nondecreasing path from u to v passing through s. If no $u \to v$ nondecreasing path passes through s, then return $-\infty$ for u, v.

Recall that a *nondecreasing* path is a path whose consecutive edge weights form a nondecreasing sequence.

Problem 1

The *equality* product of two $n \times n$ integer matrices A and B is the matrix C such that $C[i,j] = |\{k \mid A[i,k] = B[k,j]\}|$.

Recall that the dominance product of A and B is given by $(A \odot B)[i,j] = |\{k \mid A[i,k] \le B[k,j]\}|$.

(a) Suppose that the dominance product of two $n \times n$ matrices can be computed in $O(n^c)$ time. Show that the equality product of two $n \times n$ matrices can then also be computed in $O(n^c)$ time.

(b) Show that given an instance of the dominance product of $n \times n$ matrices A, B, in $O(n^2 \log n)$ time, one can convert it into an instance of dominance product of $n \times n$ matrices A', B' such that the entries of A' and B' are integers between 1 and $2n$, and the dominance product of A' and B' equals the dominance product of A and B.

(c) Suppose that the equality product of two $n \times n$ matrices can be computed in $O(n^c)$ time. Show that the dominance product of two $n \times n$ matrices can then be computed in $O(n^c \log n)$ time. (Use part (b).)

Problem 2

Consider the following matrix product of given matrices A, B, where A is Boolean and B is integer:

$$(A \odot B)[i,j] = \min\{B[k,j] \mid A[i,k] = 1\}.$$

(a) Show that for $n \times n$ matrices A and B, the above product $A \odot B$ can be computed in $O(n^{(3+\omega)/2})$ time.

(b) Consider any directed graph G for which for every vertex v the weights on the edges going out of v take at most L nonnegative integer values. (The L different values can be different for each vertex.) Then use part (a) to show that All-Pairs Shortest Paths in such an n-node G can be computed in $\tilde{O}(\sqrt{L} n^{(9+\omega)/4})$ time.

(That is, for any $\varepsilon > 0$, we can handle up to $n^{(3-\omega)/2-\varepsilon}$ distinct edge weights out of every node in truly subcubic time.)

Problem 3

Recall Seidel's Algorithm computes APSP in an unweighted undirected graph in $O(n^\omega \log n)$ time by reducing the problem to Integer Matrix Multiplication.

Show how to modify Seidel's Algorithm so that it uses Boolean Matrix Multiplication (BMM) instead of integer matrix multiplication. Specifically, perform Seidel's algorithm, but when coming out of the recursion, to figure out if the distances are odd or even, instead of performing integer matrix multiplication, perform a constant number of BMMs, thus showing that you can obtain an $O(T(n)\log n)$ algorithm for computing APSP in unweighted undirected graphs, where $T(n)$ is the running time of an algorithm that can multiply two $n \times n$ Boolean matrices.

(Hint: consider the distances obtained in the recursive step, modulo 3.)

Problem 4

In lecture, we showed how to compute the Min-Plus product of two integer matrices whose entries are from $\{-M, \dots, M\} \cup \{\infty\}$ in $\tilde{O}(Mn^\omega)$ time. In this problem, we will use this algorithm to approximate APSP.

(a) Let A, B be two $n \times n$ matrices with nonnegative integer entries bounded by n^c for some constant c. Let $C = A \star B$ be the Min-Plus product of A and B. Suppose that for some integer $M \le O(n^c)$, for every i, j, $M < C[i,j] \le 2M$. For every $\epsilon > 0$, give an $\tilde{O}(\frac{1}{\epsilon}n^\omega)$ time algorithm that given A, B, computes $\tilde{C}$ such that for every i, j,

$$C[i,j] \le \tilde{C}[i,j] \le (1+\epsilon)C[i,j].$$

(b) Let A, B be two $n \times n$ matrices with nonnegative integer entries bounded by n^c for some constant c. Let $C = A \star B$ be the Min-Plus product of A and B. For every $\epsilon > 0$, give an $\tilde{O}(\frac{1}{\epsilon}n^\omega)$ time algorithm that given A, B, computes $\tilde{C}$ such that for every i, j,

$$C[i,j] \le \tilde{C}[i,j] \le (1+\epsilon)C[i,j].$$

(c) Let G be a directed graph whose edge weights are nonnegative integers bounded by n^c for some constant c. Let $d(i,j)$ be the distance from vertex i to vertex j. For every $\epsilon > 0$, show an $\tilde{O}(\frac{1}{\epsilon}n^\omega)$ time algorithm that given G, computes $\tilde{d}$ such that for every i, j,

$$d(i,j) \le \tilde{d}(i,j) \le (1+\epsilon)d(i,j).$$