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

Below use the following definitions:

A matrix $A$ is lower triangular if $A[i,j] = 0$ whenever $j > i$. A matrix $A$ is upper triangular if $A[i,j] = 0$ whenever $j < i$.

A LU-decomposition of an $n \times n$ matrix $A$ are two $n \times n$ matrices $L$ and $U$ such that $L$ is lower triangular, $U$ is upper triangular, and $A = LU$.

A LUP decomposition of an $m \times p$ matrix $A$ consists of three matrices $L, U, P$ where $A = LUP$ and $L$ is $m \times m$ lower triangular, $U$ is $m \times p$ upper triangular, and $P$ is a $p \times p$ permutation matrix.

Problem 0

In class we showed that if one can find a triangle in an $n$-node graph in $O(n^{3-\varepsilon})$ time for $\varepsilon > 0$, then one can multiply $n \times n$ Boolean matrices in $O(n^{3-\varepsilon/3})$ time. Consider the reduction from class and show that:

(a) If one can find a triangle in $O(n^{3}/(\log n)^c)$ time, then one can also multiply Boolean matrices in $O(n^{3}/(\log n)^c)$ time.

(b) If one can detect a triangle in an $n$ node graph in $T(n)$ time such that there is some $\varepsilon > 0$ for which $T(N) \geq (1 + \varepsilon)T(N/2)$ for all $N$, then one can also find a triangle in an $n$ node graph in $O(T(n))$ time.

Problem 1

For each of the following problems, show that an $O(n^c)$ time algorithm for any $c \geq 2$ would imply an $O(n^c)$ time algorithm for multiplying two $n \times n$ matrices.

(a) Given an $n \times n$ matrix $A$, compute $A \cdot A$.

(b) Given a lower triangular matrix $A$ and an upper triangular matrix $B$, compute $A \cdot B$.

(c) Given two lower triangular matrices $A$ and $B$, compute $A \cdot B$.

Problem 2

Suppose that there is an $O(n^c)$ time algorithm (for some $c \geq 2$) that given two $n \times n$ binary matrices $A$ and $B$, can compute their product $AB$ (over the integers) in $O(n^c)$ time. Show that then there is an $O(b^2n^c)$ time algorithm that can compute the product of two $n \times n$ matrices with entries in $\{0, \ldots, 2^b\}$.

Problem 3

Does every invertible matrix have a LU-decomposition? If so, provide a proof. If not, give an example of such a matrix and prove that no LU-decomposition exists for it.

Problem 4

Suppose that one can compute the LUP decomposition of an invertible $n \times n$ matrix in $O(n^\omega)$ time where $\omega$ is the matrix multiplication exponent. (And every invertible matrix has a LUP decomposition.) Then show the following:

(a) Given an invertible $n \times n$ matrix $A$ and an $n \times 1$ vector $b$, one can solve the linear system $Ax = b$ in $O(n^\omega)$ time.
(b) Given a (possibly non-invertible) $n \times n$ matrix $A$, one can compute its determinant $\Delta$ in $O(n^\omega \log |\Delta|)$ time.

**Problem 5: BONUS**

(you do not have to solve this problem but it is worth some bonus points)

Show that if one can multiply $n \times n$ matrices in $O(n^\omega)$ time, then one can compute the LUP decomposition of an $n \times n$ invertible matrix in $O(n^\omega)$ time.