Problem 1

Recall the Tutte matrix of a graph \( G = (V, E) \) is the matrix \( T \) with

\[
T = \begin{cases} 
    0 & \text{if } (i, j) \notin E \\
    x_{ij} & \text{if } (i, j) \in E \text{ and } i < j \\
    -x_{ij} & \text{if } (i, j) \in E \text{ and } i > j
\end{cases}
\]

Above, the \( x_{ij} \) are the \(|E| \) distinct variables corresponding to the edges. Show that the rank of \( T \) is twice the size of a maximum matching of \( G \).

Hint: Use the following fact (without proof): Let \( A \) be an \( n \times n \) skew-symmetric matrix of rank \( r \). For any two sets \( S, T \in \{1, \ldots, n\} \), denote by \( A[S, T] \) the submatrix of \( A \) consisting of the rows indexed by \( S \) and the columns indexed by \( T \). Then, for any two sets \( S, T \in \{1, \ldots, n\} \) of size \( r \),

\[
\det(A[S, S]) \times \det(A[T, T]) = \det(A[S, T]) \times \det(A[T, S]).
\]

Problem 2

Let \( G = (V, E) \) be a directed graph with edge weights in \( \{1, \ldots, M\} \). The diameter of \( G \) is the quantity \( D = \max_{u,v \in V} d(u, v) \), where \( d(u, v) \) is the distance between \( u \) and \( v \) in \( G \).

Here we will show that \( D \) can be computed in \( \tilde{O}(Mn^{\omega}) \) time.

(a) Let \( N = O(M) \). Show that one can compute \( d(u, v) \) for all pairs for which \( d(u, v) \leq N \) in \( O(Mn^\omega \log n) \) time. (Hint: successive squaring.)

(b) Let \( k \leq Mn \) and \( N = O(M) \) be given integers. Show that one can compute \( d(u, v) \) for all pairs \( u, v \) for which \( d(u, v) \in [k - N, k + N] \) in \( O(Mn^\omega (\log k + \log n)) \) time.

(Hint: Use recursion and distance product on matrices with entries in \( [-O(M), O(M)] \), based on the following Fact: any \( st \)-path of weight \( w \) has a middle edge \( (u, v) \) such that \( d(s, u), d(v, t) \in [|w/2| - M, |w/2|] \).

(c) Let \( k \leq Mn \) be a given integer. Show that in \( O(Mn^\omega) \) time one can compute a matrix \( B \) such that \( B[i, j] = 1 \) if and only if \( d(i, j) \leq k \).

(Hint: use (a),(b), Boolean matrix multiplication and the middle-edge fact from above.)

(d) Conclude that using c one can compute the diameter in \( \tilde{O}(Mn^\omega) \) time overall.

Problem 3

Let \( A \) be an arithmetic algorithm if the only operations the algorithm is allowed to perform is (1) multiplication of an already computed value by a scalar, (2) multiplication of two already computed values, (3) dividing a value (or a scalar) by a nonzero value, (4) adding two values.
Let $f$ be a polynomial function over $N$ variables, $x_1, \ldots, x_N$ over some field $K$. Let $L(f)$ be the minimum number of operations that an arithmetic algorithm uses to evaluate $f$ on an arbitrary given point. Let $L(f_1, \ldots, f_k)$ be the minimum number of operations that an arithmetic algorithm uses to evaluate all of $f_1, \ldots, f_k$ on the same given (arbitrary) point. Baur-Strassen’s theorem showed that

$$L(f, \partial f / \partial x_1, \ldots, \partial f / \partial x_N) \leq 5L(f).$$

Let $F_f = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j} \right\}_{ij}$ be the set of all the second order partial derivatives of $f$. Suppose now that

$$L(f, F_f) \leq O(L(f)),$$

for all polynomials $f$. Show that then matrix multiplication is in $O(n^2)$ time.

Hint: Consider $f(u_1, \ldots, u_n, \{a_{ij}\}, \{b_{ij}\}, v_1, \ldots, v_n) = \sum_{ijk} u_i a_{ij} b_{jk} v_k$. 