CS 6550 Algorithms (Fall '10)

Instructor: Asaf Shapira

Home Assignment 4

Due date: 11/26/10

Please submit organized and well written solutions!

Problem 1. Show that if M is not a maximum matching in G, and in the process of growing the M-alternating trees of Edmonds' Algorithm we did not find an M-augmenting path, then we are guaranteed to find a blossom.

Problem 2. For a subset of vertices $U \subseteq V(G)$ we denote by $q(G-U)$ the number of connected components of $G-U$ of odd size.

- Show that for any matching M in G and any U we have $2|M| \leq |V| + |U| - q(G-U)$.
- Suppose M is a maximum matching in G and in the process of growing the M-alternating trees of Edmonds' Algorithm we did not find a blossom. Show that if we take U to be the vertices labeled "Odd" then $2|M| = |V| + |U| - q(G-U)$.
- Show that "un-shrinking" a blossom maintains the equality in the previous item and so we have the *Tutte-Berge Formula*:

$$\max_M \ 2|M| = \min_U \ (|V| + |U| - q(G-U))$$

Problem 3. Let G be a directed un-weighted graph. We say that a matrix D gives a $(1+\epsilon)$-approximation to the shortest paths of G if we have $\delta_G(u,v) \leq D(u,v) \leq (1+\epsilon)\delta_G(u,v)$ for any (ordered) pair of vertices u, v (where $\delta_G(u,v)$ is the distance from u to v in G). Show that given G and $\epsilon > 0$ we can compute a $(1+\epsilon)$-approximation matrix in time $O((n^\omega/\epsilon)\log n)$.

Problem 4. Show that Seidel's APSP algorithm can be implemented using only fast Boolean matrix multiplication. In other words, show how to eliminate the algorithm's use of fast matrix multiplication over $\mathbb{Z}$.

Problem 5. Let $x = x_0 + ix_1$ and $y = y_0 + iy_1$ be two complex numbers. Show that their product $xy = (x_0y_0 - x_1y_1) + i(x_0y_1 + x_1y_0)$ can be computed using 3 real multiplications and 5 real additions/subtractions.

- Use the same idea to show that two n-bit integers can be computed using 3 multiplication of two $\frac{n}{2}$-bit integers and some additions of n-bit integers. Note that we can do "shifts" for free.
- Use the previous item to show that two n-bit integers can be multiplied in time $O(n^{\log_2 3})$.

Problem 6. Let $TC(n)$ denote the time it takes to compute the transitive-closure of a directed n-vertex graph and $B(n)$ the time it takes to compute the boolean product of two $n \times n$ matrices. Show that $B(n/3) \leq TC(n) \leq B(n)\log n$.