CS420/520: Graph Theory with Applications to CS, Winter 2018

Homework 3

Due: Tue, 2/20/18

Homework Policy:

1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. **Solutions must be typeset.**

Readings:

(A) Jeff lecture notes on shortest paths: http://jeffe.cs.illinois.edu/teaching/algorithms/notes/21-sssp.pdf.

(B) Uri Zwick's lecture notes on all pairs shortest paths: http://www.cs.tau.ac.il/~zwick/grad-algo-13/match.pdf.

Problem 1. Let $G = (V, E)$ be a connected directed graph with non-negative edge weights, let s and t be vertices of G, and let H be a subgraph of G obtained by deleting some edges. Suppose we want to reinsert exactly one edge from G back into H, so that the shortest path from s to t in the resulting graph is as short as possible. Describe and analyze an algorithm that chooses the best edge to reinsert, in $O(E \log V)$ time.

Problem 2.

(A) A matching is maximal if it does not leave any free edge. Let M be a maximal matching and M^* be a maximum matching. Prove that $|M| \geq \frac{1}{2} \cdot |M^*|$. Then, describe an $O(E)$ time 2-approximation algorithm for computing the maximum matching (an $O(E)$ time algorithm that computes a matching of size at least 1/2 of the maximum matching) matching.

(B) Suppose the degree of all vertices is smaller than a constant Δ. Design an $O(V)$ time algorithm to find a matching M' such that $|M'| \geq \frac{2}{3}|M^*|$. What is the running time of your algorithm as a function of Δ and V?

(C) Let k be a positive integer. Modify your algorithm to find a matching M'' such that $|M''| \geq \frac{k}{k+1} \cdot |M^*|$. What is the running time of your algorithm as a function of Δ and V?

Problem 3. Describe and analyze an efficient algorithm to compute the number of shortest paths between two specified vertices s and t in a directed graph G whose edges have positive weights. [Hint: Which edges of G can lie on a shortest path from s to t?]