CS420/520: Graph Theory with Applications to CS, Winter 2014

Homework 2*

Due: Thr, Feb/20/14

Homework Policy: Each student should submit his/her own set of solutions, independently. You are allowed to discuss the homework with other students, however, you need to indicate their names in your submission. Also, you are allowed to use other sources, but you must cite every source that you use.

Problem 1. Recall that Goldberg's algorithm requires to compute shortest paths in a directed acyclic graph (DAG) with edge weights from the set $\{0, -1\}$. In this problem, we develop such an algorithm.

(a) Recall that the indegree of a vertex v is the number of edges that enter v, and the outdegree of a vertex v is the number of edges that leave v. Prove that every DAG has at least one vertex with outdegree zero, we call such a vertex a sink.

(b) Let $G = (V, E)$ be a DAG with edge weights $w : E \to \{-1, 0\}$, let v be a sink in G, and let $s \in V$. Suppose that s can reach all vertices of G by directed paths. Let $H = G/\{v\}$ be the graph we obtain by deleting v from G. Further, let $d_H(s, u)$ be the length of the shortest path from s to $u \in V/\{v\}$ in H. and let $d_G(s, u)$ be the shortest path from s to $u \in V$ in G. Write d_G as a function of d_H.

(c) Use part (b) to design and analyze an $O(m)$ time algorithm to compute d_G.

Solve two problems from the following three problems

problem 2. Show that a graph is Bipartite if and only if it does not contain any cycle of odd length (Hint: Use BFS layers.)

problem 3. Let $G = (V, E)$ be a graph and let M be a matching in G. Suppose that v is unmatched by M and that there is no augmenting path with respect to M that starts at v. Show that there exists a maximum matching M in which v is unmatched.

*Problem 3 is from Uri Zwick's lecture notes.

problem 4. Let M be a maximal matching and M^* be a maximum matching. Prove that $|M| \geq |M^*|/2$. Conclude an $O(m)$ time 2-approximation algorithm for computing the maximum matching, an $O(m)$ time algorithm that computes a matching of size at least $1/2$ of the maximum matching.