CS420/520: Graph Theory with Applications to CS, Winter 2016

Homework 3

Due: Thr, 3/10/16

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. Please typeset your solutions.

Readings:

(A) Chandra's lecture notes on ATSP: https://courses.engr.illinois.edu/cs598csc/sp2011/lectures/lecture_2.pdf.

(B) Eulerian path on wikipedia https://en.wikipedia.org/wiki/Eulerian_path.

(C) Steiner tree on wikipedia https://en.wikipedia.org/wiki/Steiner_tree_problem.

Problem 1. Consider the TSP-R problem. The input is a graph $G = (V, E)$ with non-negative edge costs as in TSP. A TSP-R tour is a minimum-cost *walk* that visits all vertices of G and returns to the starting vertex. Show that an α-approximation for Metric-TSP implies an α-approximation for TSP-R and vice-versa.

Problem 2. Let $G = (V, E)$ be a connected graph with exactly two odd degree vertices, $u, v \in V$. Therefore, G has an Eulerian *tour*. We can adapt the algorithm we saw in class to find an Eulerian tour of G. In this exercise we examine the following different algorithm.

The algorithm starts by setting $s = u$. At each step it chooses an edge incident to s whose removal would not disconnect the graph if such an edge exists. Otherwise, it picks any remaining edge incident to s. Then, it updates s to be the other endpoint of the chosen edge, and removes this edge from G. The algorithm finishes when there is no more edge to take.

(a) Prove that this algorithm finds an Eulerian tour of G.

(b) How would you implement this algorithm, and what would be the running time?

Problem 3. Let $G = (V, E)$ be a graph, and let $T \subseteq V$ be a set of terminals. Design an $O(2^{2|T|}|V|^2)$ time exact algorithm to compute the minimum Steiner tree of T. (Hint: for any $v \in V$ and any $X \in T$, let $S(v, X)$ be the minimum Steiner tree of X that contains v. Find a recursion for $S(\cdot, \cdot)$, and turn it into a dynamic programming.)

Problem 4.

(a) Let T be a tree with maximum degree three. Show that there is an edge T whose removal splits it into two subtrees T_1 and T_2 such that $|T_1|, |T_2| \geq 1/4 \cdot |T|$.

(b) Let G be a plane triangulation, a planar graph in which every face is incident to exactly three edges. Show there is a cycle of G that contains at least $1/4$ and at most $3/4$ of the faces of G. (Hint: look at the dual graph, and use (a).)