15-750 Graduate Algorithms

Homework # 5 **due: April 5, 2001**

Groundrules:

- Homeworks will generally consist of easier *exercises* and harder *problems*. You should do the exercises by yourself, but you may work with a friend or two on the problems if you want — just write down who you worked with. One exception: no fair working with someone who has already figured out (or already knows) the answer. Of course, your writeup should be in your own words.

- If you've seen a problem before (sometimes we will give problems that are "famous"), then say that in your solution (it won't affect your score, we just want to know). Also, if you use any sources other than the book or those on the web page, write that down too. It's fine to look up a complicated sum or inequality, but don't look up an entire solution.

Exercises:

Given a weighted undirected graph G, a *traveling salesman tour* for G is the shortest tour that starts at some node, visits all the vertices of G, and then returns to the start. We will allow the tour to visit vertices multiple times (so, our goal is the shortest cycle, not the shortest simple cycle). This version of the TSP that allows vertices to be visited multiple times is sometimes called the *metric* TSP problem, because we can think of there being an implicit complete graph H defined over the nodes of G, where the length of edge (u, v) in H is the length of the shortest path between u and v in G. (By construction, edge lengths in H satisfy the triangle inequality, so H is a metric.)

1. Briefly (at most a few sentences): prove that we can get a factor of 2 approximation to the TSP by finding a minimum spanning tree T for H and then performing a depth-first traversal of T.

2. We know that T must have an even number of nodes of odd degree (only considering the edges in the tree). Why?

3. Let M be a minimum-cost perfect matching (in H) between the nodes of odd degree in T. I.e., if there are $2k$ nodes of odd degree in T, then M will consist of k edges no two of which share an endpoint. Prove that the total length of edges in M is at most one-half the length of the optimal TSP tour.

4. We have now proven that we can get a 1.5 approximation to the TSP if we can take an Eulerian tour on the union of T and M (a tour that traverses each edge in $T \cup M$ exactly once). Prove that such a tour exists and can be found efficiently.

The above algorithm is due to Christofides [1976]. Extra credit and PhD thesis: Find an algorithm that achieves a 1.49 approximation ratio.

Problems:

5. Scheduling Taxicabs. There are N locations (points) in a city. The distances between the points are given by an N by N distance matrix $\{d_{ij}\}$, where d_{ij} is the distance between points i and j.

   There are k taxicabs in this city, all of which start out at a particular location z. A sequence of n requests are to be satisfied by the taxicabs. The ith request consists of a starting point s_i and a destination point t_i. To satisfy a request, one of the taxicabs drives from wherever it is to s_i, (picks up the passenger) then drives from s_i to t_i (and lets out the passenger), and waits there until its ordered to satisfy another request. The requests must be processed in their order of occurrence, and a taxicab may only move to get to a request it is about to satisfy, or from the start to the destination point of a request.

   Give an algorithm that takes [the complete sequence of requests, the distance matrix, and the number of taxicabs, k], and finds a solution that minimizes the total distance moved by all taxicabs. The algorithm should have running time polynomial in k, n, and N. Explain why your algorithm works. (Hint: go with the flow).

6. Here is a greedy algorithm for the vertex cover problem: pick the vertex of highest degree and put it into your cover. Then, delete all edges that have been covered and repeat. Come up with a counterexample showing that, strangely enough, this does *not* guarantee a factor of 2 approximation ratio.