CS261: Exercise Set #6

For the week of May 4–8, 2015

Instructions:

(1) *Do not turn anything in.*

(2) The course staff is happy to discuss the solutions of these exercises with you in office hours or on Piazza.

(3) While these exercises are certainly not trivial, you should be able to complete them on your own (perhaps after consulting with the course staff or a friend for hints).

Exercise 26

Recall Graham's algorithm from Lecture #11: given a parameter m (the number of machines) and n jobs with processing times $p_1, \ldots, p_n$, schedule the jobs one by one, assigning the current job j to the machine that currently has the smallest load. We proved that, for every problem instance and every order of processing the jobs, the schedule produced by this algorithm has makespan (i.e., maximum machine load) at most twice the minimum possible.

Show that for every constant $c < 2$, there exists an instance for which the schedule produced by Graham's algorithm has makespan more than c times the minimum possible.

[Hint: Your bad instances will need to grow larger as c approaches 2.]

Exercise 27

Consider the following variant of the traveling salesman problem (TSP). The input is an undirected complete graph with edge costs. These edge costs need *not* satisfy the triangle inequality. The desired output is the minimum-cost cycle, not necessarily simple, that visits every vertex *at least* once.

Show how to convert a polynomial-time α-approximation algorithm for the metric TSP problem into a polynomial-time α-approximation algorithm for this (non-metric) TSP problem with repeated visits allowed.

[Hint: Define a metric TSP instance where the edges represent paths in the original (non-metric) instance.]

Exercise 28

Let $G = (V, E)$ be an undirected graph that is connected and Eulerian (i.e., all vertices have even degree). Show that G admits an Euler tour — a (not necessarily simple) cycle that uses every edge exactly once. Can you turn your proof into an $O(m)$-time algorithm, where $m = |E|$?

[Hint: Induction on $|E|$.]

Exercise 29

Recall the MST heuristic for the metric TSP problem — in Lecture #12, we showed that this is a 2-approximation algorithm. Show that, for every constant $c < 2$, there is an instance of the metric TSP problem such that the MST heuristic returns a tour with cost more than c times the minimum possible.

Exercise 30

In the *Steiner tree* problem, the input is an undirected graph $G = (V, E)$ with nonnegative edge costs, and a set $R \subseteq V$ of "terminals" or "required vertices." The goal is to output the minimum-cost set F of edges such that the subgraph (V, F) includes a u-v path for every $u, v \in R$. (MST is the special case where $R = V$.)

In lecture we briefly mentioned the *metric* special case of the problem, where the graph G is complete and the edge costs satisfy the triangle inequality. (I.e., for every triple $u, v, w \in V$, $c_{uw} \leq c_{uv} + c_{vw}$.) Show how to convert a polynomial-time α-approximation algorithm for the metric Steiner tree problem into one for the general Steiner tree problem.[1]

[Hint: See Exercise 27.]

[1] This extends the 2-approximation mentioned in lecture to the general Steiner tree problem.