Introduction to Algorithms	Problem Set 4
CS 482, Spring 2008	Due Friday, February 29

(1) Solve Chapter 6, Exercise 2 in Kleinberg & Tardos.

(2) After graduating with a degree in CS from Cornell, you've decided to put your training in algorithms to good use by becoming a freelance taxi driver. You operate your cab in a city consisting of a finite set of locations X, and you have a database which reports the amount of time required to drive between any two locations. Your job is to serve requests from people who ask to be picked up at a specific location x_i at a specific time t_i, and dropped off at another location y_i. A set of requests is *feasible* if it is possible for a single taxi cab to serve all of them. In other words, the cab can't serve two requests simultaneously, and it also needs to have enough time to drive from the drop-off point of one request to the pick-up point of the next one in order to arrive there at (or before) the requested pick-up time.

Given an input consisting of the set X, the distance matrix $D(x, y)$, and a finite sequence of requests (x_i, y_i, t_i), design an efficient algorithm to compute the *maximum number of requests* that can be served feasibly.

(3) You're consulting for the state highway authority on a project to determine where they should place speed limit signs on a new highway. Assume that the highway has a length of L miles, and that points on the highway are identified by non-negative numbers d representing their distance from the west endpoint. The highway authority wants to enforce two types of speed constraints: a maximum speed of $s_{\max}$ miles per hour, which applies to the entire length of the highway, and local speed limits which can be expressed by ordered pairs of integers (s_i, d_i) expressing the constraint that the cars should be moving at a speed *no faster than* s_i miles per hour when passing through point d_i on the highway. There are n such constraints, numbered (s_1, d_1) through (s_n, d_n).

A speed limit sign is designated by a pair of integers (s, d), meaning that at point d on the highway there is a sign telling drivers that their speed must be at most s when traveling between d and d', where $d' > d$ is the position of the next speed limit sign on the highway, or $d' = L$ if there are no speed limit signs after d. For reasons that make sense only to bureaucrats and CS 482 professors, the highway authority has decided that they can only use k speed limit signs, where k is a positive number less than n. A collection of speed limit signs is *valid* if:

1. There are at most k signs.
2. One of the signs is at the highway's starting point, i.e. at 0.
3. Drivers obeying the posted speed limits will always obey the maximum speed constraint $s_{\max}$ as well as all of the local constraints $(s_1, d_1), \ldots, (s_n, d_n)$.
4. None of the signs is located at a point in the set $\{d_1, \ldots, d_n\}$.

Subject to these conditions, the highway authority wants to allow drivers to get from 0 to L as rapidly as possible while obeying the speed limits. Design an efficient algorithm that computes the minimum time T such that there is a valid collection of speed limit signs allowing drivers to get from 0 to L in T hours.