Introduction to Algorithms	Problem Set 4
CS 4820, Spring 2018	Due 11:59pm Friday, February 23

Due to the February break, this assignment has only one problem. Furthermore, the deadline is Friday night rather than the usual Thursday night deadline.

Hand in your solution electronically using CMS. Collaboration is encouraged while solving the problems, but:

1. **list the names of those with whom you collaborated;**
2. **you must write up the solutions in your own words;**
3. **you must write your own code.**

Remember that when a problem asks you to design an algorithm, you must also prove the algorithm's correctness and analyze its running time. The running time must be bounded by a polynomial function of the input size *unless otherwise specified.*

(1) *(10 points)*
You've been hired to consult for a local ride sharing service. The map of Ithaca is divided into a finite set of locations, X, and you have a dataset which reports the amount of time required to drive between any two locations, and the fare charged for driving a customer between those two locations. (In other words, you have matrices D and F such that $D(x, y)$ denotes the amount of time required to drive from x to y, and $F(x, y)$ denotes the fare to be charged. For the purpose of this homework problem, we will make the unrealistic assumption that $D(x, y)$ is known *precisely*, with no uncertainty.)

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 vehicle to serve all of them. In other words, the vehicle 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 set of requests (x_i, y_i, t_i), design an efficient algorithm to compute the *maximum total fare* that can be charged for serving a feasible subset of the requests.