Introduction to Algorithms	**Problem Set 2**
CS 482, Spring 2008	**Due Friday, February 8**

Please hand in each problem on a separate sheet with your name on each.

Related reading for this week's problem set: Sections 4.1, 4.2, 4.5, 4.6 of Kleinberg & Tardos.

(1) Fans of the social bookmarking site del.u.ge have been unable to reach the site since last night, when its datacenter was ironically wiped out by a flood. You are in charge of coordinating the emergency response for the company that owns del.u.ge and several other websites that were also hosted in the same datacenter. This involves restoring service using backup servers located in a different datacenter unaffected by the disaster. Unfortunately, bringing any one of these websites on line is not a simple matter of flipping a switch: it requires hours of software and hardware installation, configuration, and testing. For each of the affected websites, you have been given estimates of:

- the number of hours required to restore service for that website;
- the number of dollars that the company is losing each hour that the website remains offline.

The crew can work on restoring only one website at a time. Design an efficient algorithm that computes the optimal order in which to restore the websites, to minimize the amount of lost revenue. (You may assume that for each website, the company loses a *positive* number of dollars for each hour that it's offline. Bonus question: debate whether this assumption is realistic.)

Example: Suppose there are three websites with the following parameters.

Site	**Hours to restore**	**\$/hr lost**
del.u.ge	10	\$100,000
greedy.algo.rit.hm	4	\$50,000
cs482.com	1	\$10

The ordering del.u.ge, then greedy.algo.rit.hm, then cs482.com, results in losing

$$\$1{,}700{,}150 = (\$100\text{K} \times 10) + (\$50\text{K} \times 14) + (\$10 \times 15)$$

The ordering cs482.com, then greedy.algo.rit.hm, then del.u.ge, results in losing

$$\$1{,}750{,}010 = (\$10 \times 1) + (\$50\text{K} \times 5) + (\$100\text{K} \times 15).$$

The optimal ordering is greedy.algo.rit.hm, then del.u.ge, then cs482.com, which results in losing

$$\$1{,}600{,}150 = (\$50\text{K} \times 4) + (\$100\text{K} \times 14) + (\$10 \times 15).$$

(2) In class, we used an exchange argument to prove the correctness of Kruskal's and Prim's algorithms for the minimum spanning tree problem. This exercise explores the question of whether we could have used a "greedy stays ahead" argument instead. Of course, there are usually several different ways to formalize the intuition that the greedy algorithm stays ahead. In statements (a) and (b) below, we have chosen one natural way of formalizing this for Kruskal's algorithm, and one for Prim's.

(a) The first k edges chosen by Kruskal's algorithm have the following property: there is no cheaper acyclic subgraph of G with k edges.

(b) The first k edges chosen by Prim's algorithm (starting from some "root node" r) have the following property: there is no cheaper connected subgraph of G containing the node r along with k other nodes.

Prove one of these statements and give a counterexample to the other one.

(3) Solve Chapter 4, Exercise 19 from Kleinberg & Tardos.