Introduction to Algorithms	**Problem Set 1**
CS 482, Spring 2008	**Due Friday, February 1**

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

Hand in an additional sheet with your signature, declaring whether you do or don't give permission for your homework to be returned in class. If you don't want your homework to be returned in class, please mark "HOLD" on each page. These homeworks will be returned in Upson 360.

(0) [**Not to be handed in or graded.**] Join the mailing list for CS 482. Visit

`https://lists.cs.cornell.edu/mailman/listinfo/cs482-l`

and follow the instructions under the heading "Subscribing to Cs482-l".

(1) Solve Chapter 4, Exercise 5 from Kleinberg & Tardos.

(2) Consider the following scenario: n computer science students get flown out to the Pacific Northwest for a day of interviews at a large software company. The interviews are organized as follows. There are m time slots during the day, and n interviewers, where $m > n$. Each student S has a fixed *schedule* which gives, for each of the n interviewers, the time slot in which S meets with that interviewer. This, in turn, defines a schedule for each interviewer I, giving the time slots in which I meets each student. The schedules have the property that

- each student sees each interviewer exactly once,
- no two students see the same interviewer in the same time slot, and
- no two interviewers see the same student in the same time slot.

Now, the interviewers decide that a full day of interviews like this seems pretty tedious, so they come up with the following scheme. Each interviewer I will pick a *distinct* student S. At the end of I's scheduled meeting with S, I will take S out for coffee at one of the numerous local cafes, and they'll both blow off the entire rest of the day drinking espresso and watching it rain.

Specifically, the plan is for each interviewer I, and his or her chosen student S, to *truncate* their schedules at the time of their meeting; in other words, they will follow their original schedules up to the time slot of this meeting, and then they will cancel all their meetings for the entire rest of the day.

The crucial thing is, the interviewers want to plan this cooperatively so as to avoid the following *bad situation*: some student S whose schedule has not yet been truncated (and so is still following his/her original schedule) shows up for an interview with an interviewer who's already left for the day.

Give an efficient algorithm to arrange the coordinated departures of the interviewers and students so that this scheme works out and the *bad situation* described above does not happen.

Example: Suppose $n = 2$ and $m = 4$; there are students S_1 and S_2, and interviewers I_1 and I_2. Suppose S_1 is scheduled to meet I_1 in slot 1 and meet I_2 in slot 3; S_2 is scheduled to meet I_1 in slot 2 and I_2 in slot 4. Then the only solution would be to have I_1 leave with S_2 and I_2 leave with S_1; if we scheduled I_1 to leave with S_1, then we'd have a bad situation in which I_1 has already left the building at the end of the first slot, but S_2 still shows up for a meeting with I_1 at the beginning of the second slot.

(3) In this problem, we will investigate the potential impact of a malicious hacker attempting to sabotage the outcome of the Gale-Shapley stable matching algorithm. Assume that the hacker knows the preference list of every man and woman, and has the power to execute the following type of attack: it can intercept the preference list of a single individual (man or woman) and substitute an arbitrary preference list chosen by the hacker. The Gale-Shapley algorithm is then executed on this modified input. Let $\mathcal{M}_{\text{hack}}$ denote the resulting matching, and let $\mathcal{M}_{\text{orig}}$ denote the matching produced by running the Gale-Shapley algorithm on the original, unmodified input. We say that the attack is a *catastrophe for men* if every man strictly prefers his mate in $\mathcal{M}_{\text{orig}}$ to his mate in $\mathcal{M}_{\text{hack}}$, and we say it is a *catastrophe for women* if every woman strictly prefers her mate in $\mathcal{M}_{\text{orig}}$ to her mate in $\mathcal{M}_{\text{hack}}$. (In the preceding sentence, "strictly prefers" refers to the person's actual preferences, not the fake preference list which may have been substituted by the hacker.)

For every integer $n > 1$, prove the following:

(a) There exists a stable matching instance with n men and n women such that a hacker can cause a catastrophe for men by changing one man's preference list.

(b) There exists a stable matching instance with n men and n women such that a hacker can cause a catastrophe for women by changing one man's preference list.