CS 4800: Algorithms and Data
Due **Thursday, September 13, 2012** in class.

Homework 1

Exercise 1 (2 points)

Problem: Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

> *True or false? In every instance of the Stable Matching Problem, there is a stable matching containing a pair* (m,w) *such that* m *is ranked first on the preference list of* w *and* w *is ranked first on the preference list of* m*.*

Exercise 2 (2 points)

Problem: Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

> *True or false? Consider an instance of the Stable Matching Problem in which there exists a man* m *and a woman* w *such that* m *is ranked first on the preference list of* w *and* w *is ranked first on the preference list of* m*. Then in every stable matching* S *for this instance, the pair* (m,w) *belongs to* S*.*

Exercise 3 (5 points)

Problem: Gale and Shapley published their paper on the Stable Matching Problem in 1962, but a version of their algorithm had already been in use for ten years by the National Resident Matching Program for the problem of assigning medical residents to hospitals.

Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.)

We say that an assignment of students to hospitals is *stable* if neither of the following situations arises.

- First type of instability: there are students s and s', and a hospital h, so that

- s is assigned to h, and
- s' is assigned to no hospital, and
- h prefers s' to s.

- Second type of instability: there are students s and s', and hospitals h and h', so that
  - s is assigned to h, and
  - s' is assigned to h', and
  - h prefers s' to s, and
  - s' prefers h to h'.

So we basically have the Stable Matching Problem, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students.

Show that there is always a stable assignment of students to hospitals, and give an algorithm to find one.

Exercise 4 (5 points)

Problem: Peripatetic Shipping Lines, Inc., is a shipping company that owns n ships and provides service to n ports. Each of its ships has a *schedule* that says, for each day of the month, which of the ports it's currently visiting or whether it's out at sea. (You can assume the "month" here has m days, for some $m > n$.) Each ship visits each port for exactly one day during the month. For safety reasons, PSL Inc. has the following strict requirement:

(†) No two ships can be in the same port on the same day.

The company wants to perform maintenance on all the ships this month, via the following scheme. They want to *truncate* each ship's schedule: for each ship S_i, there will be some day when it arrives in its scheduled port and simply remains there for the rest of the month (for maintenance). This means that S_i will not visit the remaining ports on its schedule (if any) that month, but this is okay. So the *truncation* of S_i's schedule will simply consist of its original schedule up to a certain specified day on which it is in a port P; the remainder of the truncated schedule simply has it remain in port P.

Now the company's question to you is the following: Given the schedule for each ship, find a truncation of each so that condition (†) continues to hold: no two ships are ever in the same port on the same day.

Show that such a set of truncations can always be found, and give an algorithm to find them.

Example: Suppose we have two ships and two ports, and the "month" has four days. Suppose the first ship's schedule is

port P_1; at sea; port P_2; at sea

and the second ship's schedule is

at sea; port P_1; at sea; port P_2

Then the (only) way to choose truncations would be to have the first ship remain in port P_2 starting on day 3, and have the second ship remain in port P_1 starting on day 2.