Problems (solutions to be submitted electronically on Blackboard and on paper in class)

1. There are $n$ teaching assistants (TAs) and $n$ courses; one teaching assistant is to be assigned to each course. Each course ranks all of the TAs, and each TA ranks all courses, from most to least desirable.

My friend has proposed a method to find a stable assignment, as follows. First randomly assign a different course to each TA. Then, as long as there is an instability of the form TA $T_1$ is assigned course $C_2$ but prefers $C_1$, and course $C_1$ is assigned TA $T_2$ but prefers $T_1$, change the assignment to replace $C_1 - T_2$ and $C_2 - T_1$ by $C_1 - T_1$ and $C_2 - T_2$. Repeat until there are no unstable pairs.

(a) Is this an algorithm? (That is, does it terminate?) Explain.

(b) If it terminates, bound the number of number of instabilities that may be fixed as a function of $n$ – use $O()$ notation. If it does not, devise a method to resolve instabilities one by one so that it does terminate.

(c) Outline a data structure to determine which pairs are unstable throughout the method. How do you update it after a change in the assignment?

2. We got tired of trying to understand our friend’s method in Question 1, so we looked in an algorithms book and found the Gale-Shapley propose-and-reject method. So we have made an assignment using the GS algorithm. But now one of the professors complains that his course’s preferences were not recorded correctly, and he wants the whole assignment redone so that it is stable with his corrected preferences. (All of the other courses maintain their original preferences.)

(a) Devise a modification of the GS algorithm to handle such a change in one course’s preferences. Explain.

(b) Provide the best bound that you can on the number of offers to be made to correct the assignment.

3. Still on the TA assignment problem, we decide to try some evil social engineering, and see if we can make everyone unhappy.

(a) Give an example of preference lists and an assignment for which every TA and every course is in an unstable pair.

(b) Is it possible, for certain preference lists, that every assignment is stable? Explain. Clarification: If you answer yes, for which values of $n$ can this occur?

4. Looking at the amount of work that I have to get done this semester, it has become clear that I cannot get it all done. So I have to make some hard decisions. I suppose that I have $n$ tasks, each with a start date and a due date; each task gets me a reward of 10 points if I complete it, and I lose a point if I do not. Unfortunately if I decide to do a task, I can only work on that single task from the beginning of the start date to the day before the due date. I need to get as many points as I can.

(a) If the start dates and due dates for the $n$ tasks are known in advance, outline an efficient algorithm for selecting the tasks to do.

(b) Suppose that the number of tasks, and the start and due dates for each, are not known in advance. Rather you are told about each task at the beginning of the start date for that task. If,
once you have started a task you must complete it, show that you cannot efficiently maximize your number of points.

(c) Same scenario as (b) but now suppose that you can abandon the task that you are working on and switch to a new task. Can you efficiently maximize your number of points? Explain.

5 Again I have $n$ tasks, each with a start date and a due date, and all information is known in advance; each task gets me a reward of 10 points if I complete it, and I lose a point if I do not. Fortunately, although I must work on a task from the beginning of its start date to the day before its due date if I decide to do the task, I can work on six tasks at once! I need to get as many points as I can.

I use my method in 4(a) six times. Each time I examine the remaining tasks and select a subset that earns me the most points; then I remove those tasks from the list of remaining tasks before the next iteration. Will I earn the maximum possible number of points? Explain.