Problems for Recitation 7

1 A Protocol for College Admission

Next, we are going to talk about a generalization of the stable marriage problem. Recall that we have some horses and we'd like to pair them with stables so that there is no incentive for two horses to swap stables. Oh wait, that's a different problem.

The problem we're going to talk about is a generalization of the one done in lecture. In the new problem, there are N students $s_1, s_2, \ldots, s_N$ and M universities $u_1, u_2, \ldots, u_M$. University u_i has n_i slots for students, and we're guaranteed that $\sum_{i=1}^{M} n_i = N$. Each student ranks all universities (no ties) and each university ranks all students (no ties).

Design an algorithm to assign students to universities with the following properties

1. Every student is assigned to one university.
2. $\forall i$, u_i gets assigned n_i students.
3. There does not exist s_i, s_j, u_k, u_ℓ where s_i is assigned to u_k, s_j is assigned to u_ℓ, s_j prefers u_k to u_ℓ, and u_k prefers s_j to s_i.
4. It is student-optimal. This means that of all possible assignments satisfying the first three properties, the students get their top choice of university amongst these assignments.

The algorithm will be a slight modification of the mating algorithm given in lecture. For your convenience, we have provided a copy of the mating algorithm on the next page.

Each Day:

- Morning:
  - Each girl stands on her balcony
  - Each boy stands under the balcony of his favorite girl whom he has not yet crossed off his list and serenades. If there are no girls left on his list, he stays home and does 6.042 homework.
- Afternoon:
  - Girls who have at least one suitor say to their favorite from among the suitors that day: "Maybe, come back tomorrow."
  - To the others, they say "No, I will never marry you!"
- Evening:
  - Any boy who hears "No" crosses that girl off his list.

Termination Condition: If there is a day when every girl has at most one suitor, we stop and each girl marries her current suitor (if any).

1. Before we can say anything about our algorithm, we need to show that it terminates. Show that the algorithm terminates after $NM + 1$ days.

2. Next, we will show that the four properties stated earlier are true of our algorithm. To start, let's show the following: if during some day a university u_j has at least n_j applicants, then when the algorithm terminates it accepts exactly n_j students.

3. Next, show that every student is assigned to one university.

4. Next, show that for all i, u_i gets assigned n_i students.

5. Before continuing, we need to establish the following property. Suppose that on some day a university u_j has at least n_j applicants. Define the *rank* of an applicant s_i with respect to a university u_j as s_i's location on u_j's preference list. So, for example, u_j's favorite student has rank 1. Show that the rank of u_j's least favorite applicant that it says "Maybe, ..." to cannot decrease (e.g., going from 1000 to 1005 is decreasing) on any future day. Note that u_j's least favorite applicant might change from one day to the next.

6. Next, show there does not exist s_i, s_j, u_k, and u_ℓ where s_i is assigned to u_k, s_j is assigned to u_ℓ, s_j prefers u_k to u_ℓ, and u_k prefers s_j to s_i. Note that this is analogous to a "rogue couple" considered in lecture.

7. Finally, we show in a very precise sense that this algorithm is *student-optimal*. As in lecture, define the *realm of possibility* of a student to be the set of all universities u, for which there exists some assignment satisfying the first three properties above, in which the student is assigned to u. Of all universities in the realm of possibility of a student we say that the student's favorite is *optimal* for that student.

   Show that each student is assigned to its optimal university.