CSE 521: Design and Analysis of Algorithms
Assignment #1
April 5, 2002
Due: Friday, April 12

Reading Assignment: Kleinberg and Tardos, Chapter 1, Sections 3.1-3.2

Problems:

1. Gale and Shapley published their paper on the stable marriage 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 marriage problem from class, 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 efficient algorithm to find one. The input size is $\Theta(mn)$; ideally, you would like to find an algorithm with this running time.

2. We can think about a different generalization of the stable matching problem, in which certain man-woman pairs are explicitly *forbidden*. In the case of employers and applicants, picture that certain applicants simply lack the necessary qualifications or degree; and so they cannot be employed at certain companies, however desirable they may seem. Concretely, we have a set M of n men, a set W of n women, and a set $F \subseteq M \times W$ of pairs who are simply *not allowed* to get married. Each man m ranks all the women w for which $(m, w) \notin F$, and each woman w' ranks all the men m' for which $(m', w') \notin F$.

   In this more general setting, we say that a matching S is *stable* if it does not exhibit any of the following types of instability.

   (i) There are two pairs (m, w) and (m', w') in S with the property that m prefers w' to w, and w' prefers m to m'. *(The usual kind of instability.)*

   (ii) There is a pair $(m, w) \in S$, and a man m', so that m' is not part of any pair in the matching, $(m', w) \notin F$, and w prefers m' to m. *(A single man is more desirable and not forbidden.)*

   (ii′) There is a pair $(m, w) \in S$, and a woman w', so that w' is not part of any pair in the matching, $(m, w') \notin F$, and m prefers w' to w. *(A single woman is more desirable and not forbidden.)*

   (iii) There is a man m and a woman w, neither of which is part of any pair in the matching, so that $(m, w) \notin F$. *(There are two single people with nothing preventing them from getting married to each other.)*

   Note that under these more general definitions, a stable matching need not be a perfect matching.

   Now we can ask: for every set of preference lists and every set of forbidden pairs, is there always a stable matching? Resolve this question by doing one of the following two things: (a) Giving an algorithm that, for any set of preference lists and forbidden pairs, produces a stable matching; or (b) Giving an example of a set of preference lists and forbidden pairs for which there is no stable matching.

3. For this problem, we will explore the issue of *truthfulness* in the stable matching problem, and specifically in the Gale-Shapley algorithm. The basic question is: Can a man or a woman end up better off by lying about his or her preferences? More concretely, we suppose each participant has a true preference order. Now consider a woman w. Suppose w prefers man m to m', but both m and m' are low on her list of preferences. Can it be the case that by switching the order of m and m' on her list of preferences (i.e., by falsely claiming that she prefers m' to m) and running the algorithm with this false preference list, w will end up with a man m'' that she truly prefers to both m and m'? Similarly, can a man end up with a better partner by falsely switching the order of women on his preference list?

   Resolve both questions by doing one of the following two things in each case:

(a) Giving a proof that, for any set of preference lists, switching the order of a pair on the list cannot improve a man's [woman's] partner in the Gale-Shapley algorithm; or

(b) Giving an example of a set of preference lists for which there is a switch that would improve the partner of a man [woman] who switched preferences.

4. Give a critique of the use of stable matching as a way of doing TA assignments in our department. Discuss strengths and weaknesses. Propose algorithmic alternatives or improvements . Please write AT MOST one page.

5. **Extra Credit:** Give feedback on chapter 1 of book. Send this by email to Anna and Gideon.