Homework 1

1. Solve Kleinberg and Tardos, Chapter 1, Exercise 1. (5pts)

False. Consider the case with *n* = 2, with two men *m*1 and *m*2 and two women *w*1 and *w*2, where *m*1 ranks *w*1 first, *w*1 ranks *m*2 first, *m*2 ranks *w*2 first, and *w*2 ranks *m*1 first. Then there does not exist a pair (*m, w*) such that *m* ranks *w* first and *w* ranks *m* first, so there does not exist a stable matching that contains (*m, w*).

Rubric (5pt):

* + 2 pt: Correctly identifies the statement is false.
  + 3 pt: Provides a correct counterexample as explanation.

1. Solve Kleinberg and Tardos, Chapter 1, Exercise 2. (5pts)

True. Suppose *S* is a stable matching where *m* and *w* are not paired with each other. Suppose that instead *m* is matched with *w′* and *w* is matched with *m′*. Then the pairing (*m, w*) is an instability with respect to *S*, since *m* prefers *w* over *w′* and *w* prefers *m* over *m′*, contradicting the stability of *S*. Thus, every stable matching must contain (*m, w*).

Rubric (5pt):

* + 2 pt: Correctly identifies the statement is true.
  + 3 pt: Provides a correct explanation.

1. Determine whether the following statement is true or false. If it is true, give an example. If it is false, give a short explanation. (5pts)

For some *n ≥* 2, there exists a set of preferences for *n* men and *n* women such that in the stable matching returned by the G-S algorithm, every woman is matched with their most preferred man, even though that man does not prefer that woman the most.

True: Consider the following set of preferences among *n* = 3 men and women:

| Men | Preferences | Women | Preferences |
| --- | --- | --- | --- |
| 1 | C > A > B | A | 1 > 2 > 3 |
| 2 | A > B > C | B | 2 > 3 > 1 |
| 3 | A > C > B | C | 3 > 1 > 2 |

Then an execution of the G-S algorithm may proceed as follows:

1. Man 1 proposes to woman *C*, who accepts.
2. Man 2 proposes to woman *A* who accepts.
3. Man 3 proposes to woman *A*, who rejects.
4. Man 3 proposes to woman *C*, who accepts, freeing man 1.
5. Man 1 proposes to woman *A*, who accepts, freeing man 2.
6. Man 2 proposes to woman *B*, who accepts.

Then the algorithm terminates with man 1, 2, and 3 matched with woman *A*, *B*, and *C*, respectively. Note that every woman is matched with their most preferred man, even though that man does not prefer that woman the most.

Rubric (5pt):

* + 2 pt: Correctly identifies the statement is true.
  + 3 pt: Provides a correct example and explanation.

1. Four students, *a*, *b*, *c*, and *d*, are rooming in a dormitory. Each student ranks the others in strict order of preference. A *roommate matching* is defined as a partition of the students into two groups of two roommates each. A roommate matching is *stable* if no two students who are not roommates prefer each other over their roommate.

Does a stable roommate matching always exist? If yes, give a proof. Otherwise, give an example of roommate preferences where no stable roommate matching exists. (8pts)

A stable matching need not exist.

Consider the following list of preferences. Note *a*, *b*, and *c* all prefer *d* the least.

* + *a* : *b* > *c* > *d*
  + *b* : *c* > *a* > *d*
  + *c* : *a* > *b* > *d*

Now, there can only be 3 sets of disjoint roommate pairs.

* + If the students are divided as (*a, b*) and (*c, d*), then (*b, c*) cause an instability, since *c* prefers *b* over *d* and *b* prefers *c*

over *a*.

* + If the students are divided as (*a, c*) and (*b, d*), then (*a, b*) cause an instability, since *b* prefers *a* over *d* and *a* prefers *b*

over *c*.

* + If the students are divided as (*a, d*) and (*b, c*), then (*a, c*) cause an instability, since *a* prefers *c* over *d* and *c* prefers *a*

over *b*.

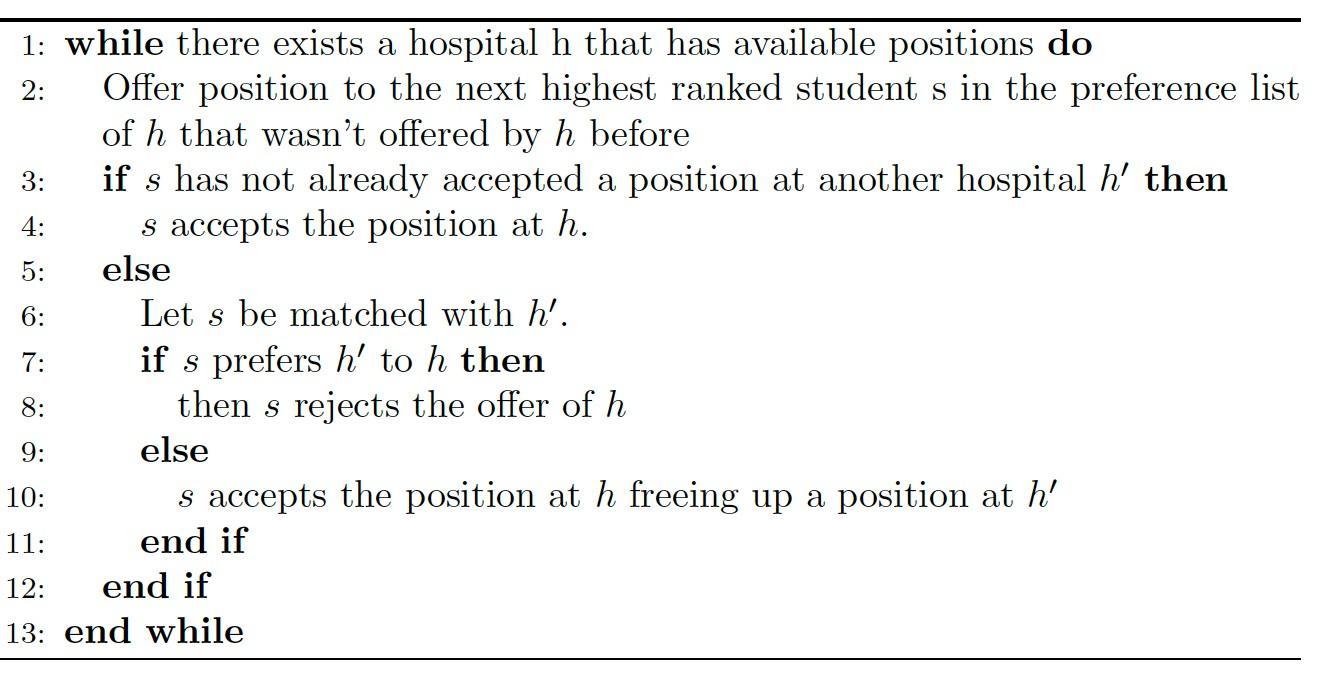
Thus every matching is unstable, and no stable matching exists with this list of preferences. Rubric (8pts):

* + 2 pts: Correctly identifies that stable matching wont always exist.
  + 8 pts: Provides a correct explanation and example.

1. Solve Kleinberg and Tardos, Chapter 1, Exercise 4. (15pts)

We will use a variation of the G-S algorithm, then show that the solution returned by this algorithm is a stable matching. In the following algorithm, we use hospitals in the place of men; and students in the place of women, with respect to the earlier version of the G-S algorithm given in Chapter 1. This algorithm terminates in *O*(*mn*) steps because each hospital offers a position to a student at most once, and in each iteration some hospital offers a position to some student.

The algorithm terminates by producing a matching *M* for any given preference list. Suppose there are *p >* 0 positions available at hospital *h*. The algorithm terminates with all of the positions filled, since, any hospital that did not fill all of its positions must have offered them to every student. Every student who rejected must be committed to some other hospital. Thus, if *h* still has available positions, it would mean total number of available positions is strictly greater than *n*, the number of students. This contradicts the assumption given, proving that all the positions get filled.



The assignment is stable. Suppose that the matching *M* produced by our adapted G-S algorithm contains one or more instabilities. If the instability was of the first type (a student *s′* was preferred over a student *s* by a hospital *h* but was not admitted), then *h* must have made an offer to *s* before *s′* who wasn’t offered, which is a contradiction because *h* prefers *s′* to *s*. Thus, the instability was not of the first type.If the instability was of the second type (there are student *s* and *s′* currently at hospitals *h* and *h′* respectively, and there’s a swap mutually beneficial to *h* and *s′*), then *h* must not have admitted *s′* when it considered it before *s*, which implies that *s′* prefers *h′* to *h*, a contradiction. Thus, the instability was not of the second type. Thus, the matching was stable. Thus at least one stable matching always exists (and it is produced by the adapted G-S algorithm as above).

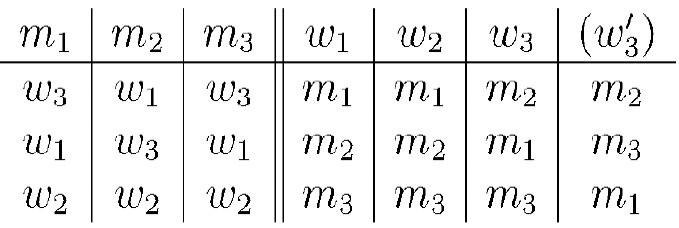
Rubric (15pts):

* + 8pts: Algorithm
    - 1pt: Loop condition (line 1)
    - 2 pts: hospitals offer next highest ranked student (line 2)
    - 2pts: case that s is free (lines 3-4)
    - 3pts: cases if s is at another h’
  + 7pts: Proof
    - 1 pt: Algorithm terminates in finite steps (optional to mention in O(mn) steps)
    - 2pts: All positions get filled
    - 2pts: Explain why no instability of first type
    - 2pts: Explain why no instability of second type

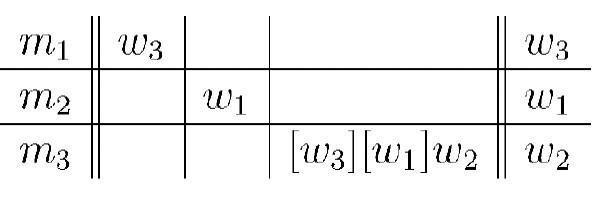
1. Solve Kleinberg and Tardos, Chapter 1, Exercise 8. (10pts)

Assume we have three men *m*1, *m*2, and *m*3 and three women *w*1, *w*2, and *w*3 with preferences as given in the table below.

Column *w*3 shows the true preferences of woman *w*3, while in column *w′* she pretends she prefers man *m*3 to *m*1.

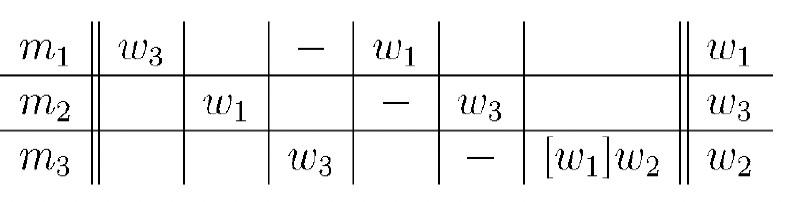


First let us consider one possible execution of the G-S algorithm with the true preference list of *w*3.



First *m*1 proposes to *w*3, then *m*2 proposes to *w*1. Then *m*3 proposes to *w*2 and *w*1 and gets rejected, finally proposes to *w*2 and is accepted. This execution forms pairs (*m*1, *w*3), (*m*2, *w*1) and (*m*3, *w*2), thus pairing *w*3 with *m*1, who is her second choice. Now consider execution of the G-S algorithm when *w*3 pretends she prefers *m*3 to *m*1 (see column *w′* ).

Then the execution might look as follows:



Man *m*1 proposes to *w*3, *m*2 to *w*1, then *m*3 to *w*3. She accepts the proposal, leaving *m*1 alone. Then *m*1 proposes to *w*1 which causes *w*1 to leave her current partner *m*2, who consequently proposes to *w*3 (and that is exactly what *w*3 prefers). Finally, the algorithm pairs up *m*3 (recently left by *w*3) and *w*2. As we see, *w*3 ends up with the man *m*2, who is her true favorite. Thus we conclude that by falsely switching order of her preferences, a woman may be able to get a more desirable partner in the G-S algorithm.

Rubric (10pts):

* 2 pts: Correctly identifies the statement is true.
* 8 pts: Provides a correct explanation and example.

1. Determine whether the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

For all *n ≥* 2, there exists a set of preferences for *n* men and *n* women such that in the stable matching returned by the G-S algorithm, every man is matched with their most preferred woman.

True: This happens so long as the most preferred woman for each man is different. Rubric (5pts):

* + 2 pts: Correctly identifies the statement is true.
  + 3 pts: Provides a correct explanation or example.

1. Consider a stable marriage problem where the set of men is given by *M* = *m*1*, m*2*, ..., mN* and the set of women is

*W* = *w*1*, w*2*, ..., wN* . Consider their preference lists to have the following properties:

*∀wi ∈ W* : *wi* prefers *mi* over *mj ∀j > i*

*∀mi ∈ M* : *mi* prefers *wi* over *wj ∀j > i*

Prove that a unique stable matching exists for this problem. Note: the *∀* symbol means “for all”. (12pts)

We will prove that matching *S* where *mi* is matched to *wi* for all *i* = 1*,* 2*, · · · , N* is the unique stable matching in this case. We will use the notation *S*(*a*) = *b* to denote that in matching *S*, *a* is matched to *b*. It is evident that S is a matching because every man is paired with exactly one woman and vice versa. If any man *mj* prefers *wk* to *wj* where *k < j*, then such

a higher ranked woman *wk* prefers her current partner to *mj*. Thus, there are no instabilities and *S* is a stable matching.

Now let’s prove that this stable matching is unique. By way of contradiction, let’s assume that another stable matching *S′*, which is different from *S*, exists. Therefore, there must exist some *i* for which *S′*(*wi*) = *mk*, *k ̸*= *i*. Let *x* be the minimum value of such an *i*. Similarly, there must exist some *j* for which *S′*(*mj*) = *wl*, *j ̸*= *l*. Let *y* be the minimum value of such

a *j*. Since *S′*(*wi*) = *mi* for all *i < x*, and *S′*(*mj*) = *wj* for all *j < y*, *x* = *y*. *S′*(*wx*) = *mk* implies *x < k*. Similarly,

*S′*(*my*) = *wl* implies that *y* = *x < l*. Given the preference lists, *my* = *mx* prefers *wx* to *wl*, and *wx* prefers *mx* to *mk*. This is an instability and hence, *S*’ cannot be a stable matching.

Rubric (12pts):

* Proof that stable matching exists (6pts)
* correct explanation that this stable matching is unique (6pts)

**UNGRADED PRACTICE PROBLEMS**

1. Determine whether the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

For all *n ≥* 2, there exists a set of preferences for *n* men and *n* women such that in the stable matching returned by the G-S algorithm, every woman is matched with their least preferred man.

True: One example occurs when every woman has a different least favorite man, who happens to prefer her over all other women. In this case, each man proposes to and engages their most preferred woman and the engagement is never broken.

1. Consider the G-S algorithm for *n* men and *n* women. What is the maximum number of times a man may be rejected as a function of *n*? Give an example where this happens.

The maximum number of times a woman may be rejected is *n −* 1, since no woman will be rejected by all *n* men.