

Math for CS 2015/2019 solutions to “In-Class Problems Week 9, Wed. (Session 22)”

<https://github.com/spamegg1>

November 5, 2022

Contents

1	Problem 1	1
1.1	(a)	2
1.2	(b)	2
2	Problem 2	2
2.1	(a)	3
2.2	(b)	3
2.3	(c)	3
2.4	(d)	3
2.5	(e)	3
3	Problem 3	4
4	Problem 4	4

1 Problem 1

Four Students want separate assignments to four VI-A Companies. Here are their preference rankings:

Student	Companies
Albert	HP, Bellcore, AT&T, Draper
Sarah	AT&T, Bellcore, Draper, HP
Tasha	HP, Draper, AT&T, Bellcore
Elizabeth	Draper, AT&T, Bellcore, HP

Company	Students
AT&T	Elizabeth, Albert, Tasha, Sarah
Bellcore	Tasha, Sarah, Albert, Elizabeth
HP	Elizabeth, Tasha, Albert, Sarah
Draper	Sarah, Elizabeth, Tasha, Albert

1.1 (a)

Use the Mating Ritual to find two stable assignments of Students to Companies.

Proof. Treat Students as Boys and the result is the following assignment:

Student	Company	Rank in the original list
Albert	Bellcore	2
Sarah	AT&T	1
Tasha	HP	1
Elizabeth	Draper	1

Treat Companies as Boys and the result is the following assignment:

Company	Student	Rank in the original list
AT&T	Albert	2
Bellcore	Sarah	2
HP	Tasha	2
Draper	Elizabeth	2

□

1.2 (b)

Describe a simple procedure to determine whether any given stable marriage problem has a unique solution, that is, only one possible stable matching.

Proof. See if the Mating Ritual with Boys as suitors yields the same solution as the algorithm with Girls as suitors. These two marriage assignments are boy-optimal and boy-pessimal, respective. Obviously, if every boy's optimal and pessimal choices are the same, then every boy has an unique choice. The solution is unique. □

2 Problem 2

Suppose that Harry is one of the boys and Alice is one of the girls in the Mating Ritual. Which of the properties below are preserved invariants? Why?

2.1 (a)

Alice is the only girl on Harry's list.

Proof. A preserved invariant; no girl will be added to Harry's list. If Alice got crossed off, there would be no one for Harry to marry. So she must remain as the sole girl on his list. Reminder: A preserved invariant need not be true all the time, as in this example. It only needs to stay true once it first becomes true. \square

2.2 (b)

There is a girl who does not have any boys serenading her.

Proof. Not preserved; a girl may not have a suitor on the first day, if, for example, she's not at the top of any boy's list, but every girl is guaranteed to have one at the end, namely, her husband. \square

2.3 (c)

If Alice is not on Harry's list, then Alice has a suitor that she prefers to Harry.

Proof. A preserved invariant; this is the basic invariant used to verify the Ritual. \square

2.4 (d)

Alice is crossed off Harry's list, and Harry prefers Alice to anyone he is serenading.

Proof. A preserved invariant; Harry crosses off the girls in his order of preference, so if Alice is crossed off, Harry likes her better than anybody that's left. \square

2.5 (e)

If Alice is on Harry's list, then she prefers Harry to any suitor she has.

Proof. Not preserved. Suppose the preferences among two couples and a third boy are: Harry: Alice, Elvira, ...

Billy: Elvira, Alice, ...

Wilfred: Elvira, ...

Alice: Billy, Harry, ...

Elvira: Wilfred, Billy, ...

The alleged invariant is true on the first day since Harry is Alice's only suitor. But Elvira rejects Billy in favor of Wilfred on the first afternoon, so on the second day, Billy and Harry are serenading Alice. Since Alice prefers Billy to Harry, the alleged invariant is no longer true, so it was not preserved. \square

3 Problem 3

A preserved invariant of the Mating Ritual is:

For every girl, G , and every boy, B , if G is crossed off B 's list, then G has a favorite suitor, and she prefers him over B .

Use the invariant to prove that the Mating Algorithm produces stable marriages. (Don't look up the proof in the Notes or slides.)

Proof. Let Brad be some boy and Jen be any girl that he is not married to on the last day of the Mating Ritual. We claim that Brad and Jen are not a rogue couple. Since Brad is an arbitrary boy, it follows that no boy is part of a rogue couple. Hence the marriages on the last day are stable.

To prove the claim, we consider two cases:

Case 1. Jen is not on Brad's list. Then by invariant P, we know that Jen prefers her husband to Brad. So she's not going to run off with Brad: the claim holds in this case.

Case 2. Otherwise, Jen is on Brad's list. But since Brad is not married to Jen, he must be choosing to serenade his wife instead of Jen, so he must prefer his wife. So he's not going to run off with Jen: the claim also holds in this case. \square

4 Problem 4

The most famous application of stable matching was in assigning graduating medical students to hospital residencies. Each hospital has a preference ranking of students, and each student has a preference ranking of hospitals, but unlike finding stable marriages between an equal number of boys and girls, hospitals generally have differing numbers of available residencies, and the total number of residencies may not equal the number of graduating students.

Explain how to adapt the Stable Matching problem with an equal number of boys and girls to this more general situation. In particular, modify the definition of stable matching so it applies in this situation, and explain how to adapt the Mating Ritual to handle it.

Proof. (excerpt from *The Stable Marriage Problem, Structures and Algorithms* by Dan Gusfield and Robert W. Irving)

Each hospital offers a place to residents, in preference order, until all of its available places are (provisionally) filled or there are no more acceptable residents on its list.

Each resident behaves exactly like a woman in the man-oriented version of the Gale-Shapley algorithm, rejecting all offers except the best received to date. To describe the extended version of this algorithm that reduces the preference lists, we introduce some suitable terminology.

We first of all assume that if a hospital h is not acceptable to a resident r , then r is deleted from h 's list, and vice versa, so that the lists are consistent from the outset. Clearly, no such pair (h, r) can block a matching.

We use the term *provisional assignment* rather than engagement in the context of residents and hospitals.

A resident who is not provisionally assigned is *free*, and

a hospital having fewer residents provisionally assigned than it has places available is *undersubscribed*.

The hospital-oriented algorithm is as follows:

assign each resident to be free;

assign each hospital to be totally unsubscribed;

while (some hospital h is undersubscribed) and

(h 's list contains a resident r not provisionally assigned to h) do

begin

$r ::=$ first such resident on h 's list;

if r is already assigned, say to h' , then

break the provisional assignment of r to h' ;

provisionally assign r to h ;

for each successor h' of h on r 's list do

remove h' and r from each other's lists

end

On termination of the hospital-oriented algorithm, each undersubscribed hospital has assigned to it all of the residents in its (reduced) list, for otherwise that hospital would have prevented termination of the loop.

Further, a fully subscribed hospital with q places has assigned to it the first q residents in its (reduced) list. Note that, as in the case of the extended Gale-Shapley algorithm for the stable marriage problem, the pair removals have the effect that no offers are immediately rejected.

Also, as with the earlier Gale-Shapley algorithm (both the original and extended versions), the nondeterministic aspect of this algorithm is of no consequence, so that the provisional assignment of residents to hospitals on termination is uniquely defined for any hospitals/residents instance. Furthermore, it turns out that the matching defined by this provisional assignment is stable and, among all stable matchings, simultaneously optimal for all the hospitals. These facts are summarized in the following lemmas and theorem.

Lemma 1. *For a given instance of the hospitals/residents problem, all possible executions of the hospital-oriented extended Gale-Shapley algorithm terminate with the same preference lists.*

Proof. Let E and E' be two different executions of the algorithm, and suppose that the pair (h, r) is deleted during E' but not during E . Suppose further that (h, r) is the first such pair deleted during E' .

The deletion of (h, r) must occur because r receives an offer from a better hospital h' . Therefore, during E , h' does not offer a place to r , and so h' must be fully subscribed with residents that it prefers to r . Therefore, during E' , one of these residents, say r' , must be deleted from the list of h' before h' makes an offer to r , contradicting our assumption. \square

We use the terminology HGS-lists, analogous to that introduced in Section 1.2.4 for the stable marriage problem, for the reduced preference lists generated by the hospital-oriented algorithm.

Lemma 2. *If the pair (h, r) is absent from the HGS-lists, then*

- (i) (h, r) cannot be a stable pair;
- (ii) r prefers all the hospitals in his HGS-list to h ;
- (iii) (h, r) cannot block any matching that is contained in the HGS-lists.

Proof. (i) Suppose that (h, r) was the first stable pair deleted during a particular execution of the algorithm, and that it was deleted when hospital h' offered a place to r , so that r must prefer h' to h .

So if $q(h')$ denotes the number of places available at hospital h' , then the number of stable pairs (h', r') such that h' strictly prefers r' to r must be less than $q(h')$. For otherwise, one of these residents r' would have to have been deleted from the list of h' before h' offered a place to r , contradicting our assumption that (h, r) was the first stable pair deleted.

Therefore, in any stable matching, h' is either undersubscribed or is assigned a resident who is inferior to r . It follows then that any supposed stable matching in which r is assigned to h is blocked by the pair (h', r) .

(ii) This is immediate, because a hospital h is removed from r 's list only when r becomes provisionally assigned to a hospital that he prefers to h .

(iii) This is an immediate consequence of (ii). \square

Theorem 1. (i) *The matching specified by the provisional assignments after the execution of the hospital-oriented algorithm is stable.*

(ii) *In this matching, a hospital h with q available places is assigned either its best q stable partners, or a set of fewer than q residents; in the latter case no other resident is assigned to h in any stable matching.*

(iii) *Each resident is assigned in this matching to his worst stable partner.*

Proof. (i) By part (iii) of the previous lemma, the matching in question cannot be blocked by any pair that is absent from the HGS-lists. Nor can it be blocked by any pair that is present in the HGS-lists, since the residents assigned to any hospital are at the head of that hospital's HGS-list.

(ii) If h 's HGS-list contains at least q residents, then, after execution of the hospital-oriented algorithm, h is matched with the first q of these. Because no stable pair is absent from the HGS-lists, it follows that these are the q best stable partners for h .

On the other hand, if h 's HGS-list contains fewer than q residents, then all of these are assigned to h in the matching, and, again because no stable pairs are absent, h has no other stable partners.

(iii) This follows also from Lemma 2, together with the fact that each resident is assigned to the last hospital in his HGS-list.

In view of this theorem, we are justified in referring to the matching generated by the hospital-oriented algorithm as the hospital-optimal and resident-pessimal stable matching. □

□