

Algorithms Solutions

Homework 1 - Chapter 1, problems 1, 2, 4

**Exercise 1.1. Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.**

True or false? In every instance of the Stable Matching Problem, there is a stable matching containing a pair  $(m, w)$  such that  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ .

False. Consider the group  $m_1, m_2, w_1$ , and  $w_2$ .

$m_1$  has the ranking:

- 1.  $w_1$
- 2.  $w_2$

$m_2$  has the ranking:

- 1.  $w_2$
- 2.  $w_1$

$w_1$  has the ranking:

- 1.  $m_2$
- 2.  $m_1$

$w_2$  has the ranking:

- 1.  $m_1$
- 2.  $m_2$

In this situation, it's impossible to have a pair  $(m, w)$ , let alone a stable matching containing a pair, such that  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ .

Consider each  $m$ ,  $m_1$  and  $m_2$ :

If  $m_1$  were in a pair with the  $w$  ranked first on the preference list of  $m_1$ , then  $m_1$  would be in a pair with  $w_1$ . However, then  $w_1$  would not be in a pair with  $m_2$ , whom is ranked first on  $w_1$ 's preference list of  $m$ .

If  $m_2$  were in a pair with the  $w$  ranked first on the preference list of  $m_2$ , then  $m_2$  would be in a pair with  $w_2$ . However, then  $w_2$  would not be in a pair with  $m_1$ , whom is ranked first on  $w_2$ 's preference list.

Therefore, in this instance, it cannot be the case that either  $m$ ,  $m_1$  or  $m_2$ , can be paired with the  $w$  ranked first on its preference list and have its "match",  $w$ , be with the  $m$  that is ranked first on its preference list. Therefore, since there is no way to include  $m$  in such a pair  $(m, w)$  such that  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ , there cannot be a stable matching containing a pair  $(m, w)$  such that  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ .

**Exercise 1.2. Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.**

True or false? Consider an instance of the Stable Matching Problem in which there exists a man  $m$  and a woman  $w$  such that  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ . Then, in every stable matching  $S$  for this instance, the pair  $(m, w)$  belongs to  $S$ .

True

**Proof by contradiction:**

Suppose that there is a stable matching  $S$  for this instance (where  $m$  is ranked first on the preference list of  $w$  and  $w$  is ranked first on the preference list of  $m$ ) with the pairs  $(m, w')$  and  $(m', w)$ . However, since  $w$  is ranked first on  $m$ 's preference list,  $m$  must prefer  $w$  to  $w'$ , and since  $m$  is ranked first on  $w$ 's preference list,  $w$  must prefer  $m$  to  $m'$ . However, this is an instability, as  $m$  and  $w$  both prefer each other to their current partners, showing that this is a contradiction and therefore the pair  $(m, w)$  must belong to  $S$ .

**Exercise 1.4. Show that there is always a stable assignment of students to hospitals, and give an algorithm to find one.**

Let the hospitals do the "selecting"

let  $H$  represent the hospitals, and  $S$  represent the students

each  $h$  in  $H$  need to fill slots, the number of slots can vary, but the total combined number of slots (includes every hospital) is less than the total number of students

each student  $s$  is free until he or she has accepted (at least temporarily), an offer from a hospital, then the student is "matched" to the hospital (but not necessarily committed; this is like the engagement in the men/women matching problem)

Initially all  $h$  in  $H$  have no slots "taken"; they are all "free" and all  $s$  in  $S$  are free and not "matched". When a slot in  $h$  is "taken", the number of free slot decreases by one. It's possible for the number of free slots in a hospital to increase if a student first accepts an offer from  $h_1$ , then gets an offer from  $h_2$  and "leaves"  $h_1$  for  $h_2$ . Now, the number of free slots  $h_1$  has has increased by 1.

Here's the algorithm:

Initially all every slot in every  $h$  in  $H$  and all  $s$  in  $S$  are free

While there is a hospital  $h$  with at least 1 free slot and that hasn't offered to every student  $S$

```
{
  Choose a hospital  $h$  with at least 1 slots

  Let  $s$  be the highest-ranked student in  $h$ 's preference list to whom  $h$  has not given an offer to

  If  $s$  is free, then  $s$  accepts the offer and the number of slots  $h$  has available decreases by 1

  Else  $s$  has previously accepted an offer from  $h'$ 
  {
    If  $s$  prefers  $h'$  to  $h$ 
    {
       $s$  rejects  $h$ 's offer and the slot remains free
    }

    Else  $s$  prefers  $h$  to  $h'$ 
    {
       $s$  accepts  $h$ 's offer and the number of slots  $h$  has available decreases by 1

      the number of slots  $h'$  has available increases by 1 since  $s$  is no longer going to work there
    }
  }
}
```

Returns the matches between every slot for every  $h$  in  $H$  and each student  $s$  in  $S$  that was matched with a slot in a hospital, along with all the free students  $S$  who were never offered slots since there are more students than slots available

Facts about the algorithm:

1. a student  $s$  is "matched" to a slot in a hospital  $h$  from the point where he/she gets the first offer, and the slots in a hospital  $h$  where  $s$  is matched get better and better in terms of  $s$ 's preference list, since  $s$  will only break out of one match and into another one if  $s$  gets offered a slot in a "better" hospital
2. The students that hospital  $h$  gives offers to get worse and worse, since  $h$  starts giving offers to students at the top of  $h$ 's preference list and continues down until  $h$  doesn't have any free slots left
3. The algorithm terminates after at most (total number of hospitals) \* (total number of slots in all hospitals) iterations

Each iteration contains a hospital making an offer to a student the hospital hasn't made an offer to before.

The maximum number of students that any hospital will make an offer to is equal to the total number of slots available at all hospitals. Show this by contradiction: Suppose that hospital  $h$  has at least 1 slot available and has made offers to  $|\text{total number of slots available}|$  students. Then  $|\text{total number of slots available}|$  students are matched with a slot in a hospital, since each student that  $h$  made an offer to either accepted  $h$ 's offer or was already matched to another hospital that  $s$  prefers. However, this is a contradiction;  $h$  has a slot available but  $|\text{total number of slots available}|$  students are matched with a slot in a hospital, implying that every slot has been taken.

Therefore, since the algorithm terminates, an assignment  $A$  of students to hospitals is generated.

Now, show that the algorithm generates a stable assignment A of students to hospitals:

Assume that there is an instability in assignment A generated by the algorithm and find a contradiction.

Assume first type of instability: s is assigned to h, s' is assigned to no hospital, and h prefers s' to s:

In this algorithm, h makes offers in order of preference, and any student s who is free accepts an offer from any hospital. Since s' isn't assigned to a hospital, s' therefore received no offers. However, had h preferred s' to s, h would have made an offer to s' before making an offer to s, so this is a contradiction to the assumption, so the first type of instability can't happen.

Assume that there is the second type of instability, there are pairs (h, s) and (h', s') in the "stable assignment", but h prefers s' to s and s' prefers h to h'.

In the algorithm, h made offers in order of preference until all slots were taken. Had h preferred s' to s, h would have made an offer to s' before making an offer to s. If no such offer was made, then h must have preferred s to s', contradicting the assumption. Had an offer been made to s' and it was rejected (or accepted then rejected later) in favor of some hospital h'' that s' must prefer to hospital h. Since h' is the eventual match for student s', h'' = h' or s' prefers h' to h''. Either way, since s' must prefer h'' to h, s' must prefer h' to h, this shows a contradiction to the assumption that s prefers h to h'.

Therefore, since neither instability can occur, assignment A generated by the algorithm must be a stable assignment of students to hospitals.