Math 416

Worksheet 2. The Gale–Shapley Algorithm

Question. Is there always a stable matching?

Theorem (Gale–Shapley, 1962[1])**.** Yes. In fact, there is an efficient algorithm to find one.

The idea of the algorithm is that each hospital makes an offer to its 1^{st}-choice student, and students with offers accept their best ones. Repeat, until all hospitals are matched.

Algorithm 1: The Gale–Shapley Algorithm

Input: sets H and S of n hospitals and n students, together with preference lists
Output: a stable matching

```
set M = ∅ ;
while there is h ∈ H unmatched that hasn't yet made an offer to every student do
    choose such a hospital h
    let s ∈ S be the highest-ranked (in h's list) student to whom h hasn't yet made an offer ;
    if s is unmatched then
        add (h, s) to M ;
    else if (h', s) ∈ M and s prefers h to h' then
        replace (h', s) in M by (h, s) ;                       // h' becomes unmatched
    else                                                       // s prefers h' to h
        h remains unmatched ;
return M.
```

Part I. *Analyzing the correctness of the algorithm*

Problem 1. Choose an example with $n = 3$ or $n = 4$, and run the algorithm on your example. Keep track of the entire execution of the algorithm, i.e., all offers made and accepted / rejected. (To make things more algorithmically interesting, make an example where there are overlapping first choices between the hospitals.)

Observation 1. Every hospital h makes offers to students in *decreasing* order of h's preferences.

Problem 2. Explain why Observation 1 is true.

Observation 2. Once $s \in S$ is matched they remain so; and if they switch, they only 'trade up' to a hospital higher on their preference list.

Problem 3. Explain why Observation 2 is true.

Observation 3. The algorithm terminates after $\leq n^2$ iterations of the `while` loop.

Problem 4. Explain why Observation 3 is true.
(*Hint:* Look at $P(t)$ = the number of offers made at or before the t^{th} iteration. Note that each iteration of the while loop represents a single *proposed offer*, so you should try to bound the number of proposed offers.)

Observation 4. The output M of the algorithm is a perfect matching.

Problem 5. Explain why Observation 4 is true.
(*Hint:* Look at the `while` loop's condition.)

[1] For which Shapley won the Nobel Prize in 2012.

Observation 5. The output M of the algorithm is a stable matching.

Problem 6. Explain why Observation 5 is true.

Problem 7. Fill in the preference arrays below (with $H = \{X, Y\}$ and $S = \{A, B\}$) in such a way that both perfect matchings are stable. Which one would the Gale–Shapley Algorithm produce?

	1^{st}	2^{nd}
X		
Y		

	1^{st}	2^{nd}
A		
B		

Part II. *Analyzing the output of the algorithm.*

There can be many stable matchings. It is interesting to ask: *Can we characterize the perfect matching from the Gale-Shapley algorithm?*

Definition. A pair $(h, s) \in H \times S$ is **valid** if there exists a stable matching M with $(h, s) \in M$. We also say that h and s are **valid partners** in this case.

For a *fixed* hospital h, we can consider all possible valid student pairings with h:

$$V_h := \{s \in S | (h, s) \text{ is valid}\}.$$

Define $\text{best}(h)$ to be the highest-ranked (according to h's preferences) student in V_h.

Problem 8. Take a second to discuss this. For example, why do we know V_h is not the empty set?

Theorem. Let $M^* = \{(h, \text{best}(h)) : h \in H\}$. Then M^* is a stable matching, and the Gale–Shapley Algorithm always produces M^*.

The matching M^* is called the **hospital-optimal** stable matching.

Problem 9. First prove that M^* is a *perfect* matching.

(*Hint:* First argue that it is enough to show that no two hospitals $h \neq h'$ satisfy $\text{best}(h) = \text{best}(h')$. Then assume that there are two hospitals of this kind and use the definition of 'best' to arrive at a contradiction.)

Problem 10. If we prove that the Gale–Shapley Algorithm always produces M^*, then have we completed the proof of the theorem?

Problem 11. Suppose toward a contradiction that the GS Algorithm does *not* produce M^*.

(a) Argue that there is a hospital h that is rejected by a student s for which (h, s) is a valid pair.

We may assume that we are considering the *first* time in the execution of the algorithm that a hospital h is rejected by its valid partner s. Since s is a valid partner of h, there is a stable matching M such that $(h, s) \in M$.

(b) What can you conclude from the fact that s rejects h?
(c) Argue that there is a hospital k such that (k, s) forms a blocking pair in M.
(d) Make sure you have a complete proof of the Theorem.
(e) Identify clearly the point in your proof where you used the fact that the rejection of h by s was the *first* (in the execution of the algorithm) rejection of a hospital by a valid partner.

Part III. *For the interested student.*

How are the students doing in M^*, the stable matching produced by GS? Not great. It turns out that M^* is the **student-pessimal** stable matching.

Definition. Let $\text{worst}(s)$ be the valid hospital for s that is lowest on s's preference list.

Corollary. $M^* = \{(\text{worst}(s), s) : s \in S\}$.

Problem 12.

(a) Why is it enough to show the inclusion $M^* \supseteq \{(\text{worst}(s), s) : s \in S\}$?

(b) Finish the proof of the Corollary.

(*Hint:* Assume toward a contradiction that there is a pair $(h, s) \in M^*$ with $h \neq \text{worst}(s)$. This means that there is a stable matching M with $(\text{worst}(s), s) \in M$. Find a blocking pair in M. Don't forget that $s = \text{best}(h)$.)

Examples and cultural diversion Fix preference lists for H and S. We define **cost** functions for a stable matching M as follows.

$$c_H(M) = \sum_{h \in H} (\text{rank in } h\text{'s preference list of its match in } M)$$

$$c_S(M) = \sum_{s \in S} (\text{rank in } s\text{'s preference list of its match in } M)$$

(The top choice of h gets rank 1, the next gets rank 2, etc.)

Problem 13. Using our characterization of the GS matching, insert *maximal* in one box and *minimal* in the other:

The GS matching $M*$ is the stable matching for which c_H is [ ]. and for which c_S is [ ].

Consider the example where the hospitals W, X, Y, Z and the students A, B, C, D have preferences as shown below.

	1st	2nd	3rd	4th
W	A	B	C	D
X	B	A	D	C
Y	C	D	A	B
Z	D	C	B	A

	1st	2nd	3rd	4th
A	Z	Y	X	W
B	Y	Z	W	X
C	X	W	Z	Y
D	W	X	Y	Z

Problem 14. Which stable matching does the GS produce? How many iterations of the `while` loop does the algorithm require before it halts?

Problem 15. This example admits many stable matchings.

(a) One is $\{WA, XB, YC, ZD\}$. Verify that its c_H is 4 and its c_S is 16.

(b) Verify that $\{WB, XD, YA, ZC\}$ is a stable matching, and find its costs c_H and c_S.

(c) Verify that $\{WD, XC, YA, ZB\}$ is a stable matching, and find its costs c_H and c_S.

In light of the hospital-optimality and student-pessimality of the GS matching, you might try to find a stable matching that doesn't favor one side or the other too much. Here are a few attempts to define this.

Definition. A stable matching is called ...

$$\left\{ \begin{array}{lll} \textbf{SH-equal} & \text{if} & |c_H(M) - c_S(M)| \\ \textbf{balanced} & \text{if} & \max\{c_H(M), c_S(M)\} \\ \textbf{egalitarian} & \text{if} & c_H(M) + c_S(M) \end{array} \right\} \text{ is minimal among all stable matchings.}$$

Finding SH-equal or balanced stable matchings is known to be NP-hard. But it's known how to find egalitarian stable matchings in polynomial time!