DS 320 Algorithms for Data Science	**Lecture #23 Worksheet**
Spring 2025	Prof. Kira Goldner and Jeffrey Considine

Stable Matching

The stable matching problem is everywhere in practice:

Its algorithmic solution is a real "killer app" with huge impact. Lloyd Shapley and Alvin Roth won the Nobel Prize in Economics in 2012 for the Gale-Shapley algorithm and their work on applying it in practice.

For now, we'll simplify NRMP Match problem to pretend that each hospital hires exactly one doctor—we're looking for a *perfect matching*—but it's easy to modify the scenario to handle hospitals hiring multiple doctors, schools admitting multiple students, etc.

We'll have as input:

- A set of n doctors $D = \{d_1, \dots, d_n\}$
- A set of n hospitals $H = \{h_1, \dots, h_n\}$.
- Each doctor $d \in D$ ranks all the hospitals, preferring h to h' if $h \succ_d h'$, that is, h is ranked higher than h' in d's ranking.
- Each hospital $h \in H$ ranks all the doctors, preferring d to d' if $d \succ_h d'$, that is, d is ranked higher than d' in h's ranking.

We seek a perfect matching M consisting of n pairs of the form $(d_i, h_j) \in D \times H$ where each member of D and each member of H appears in exactly one pair in M.

Problem with the NRMP Match: some pairs $(d, h), (d', h')$ would be proposed in the matching, but

$$h' \succ_d h \qquad \text{and} \qquad d \succ_{h'} d',$$

that is, doctor d prefers h' to h, and hospital h' prefers d to d', so doctor d and hospital h' would prefer to deviate from the assignment and form a pair together as (d, h'). We call such a pair that prefers one another to their assigned partners a *blocking pair.*

A matching M is *stable* if and only if it has no blocking pairs. We want to come up with an algorithm to find a stable matching.

Basic ideas:

- Initially, everyone is unmatched. Suppose an unmatched hospital h chooses the doctor d who ranks highest on their preference list and "proposes" to them, offering them a job. Can we declare immediately that (d, h) will be one of the pairs in our final stable matching?

- If some are engaged and some are free, what should happen next?

- How do we know when we're done?

The hospital-proposing Gale-Shapley algorithm:

1. Each hospital "proposes" to their favorite doctor on their list.
2. Each doctor who receives at least one proposal "gets engaged" to the hospital they prefer among those who propose; "rejects" the rest. Doctors with no proposals do nothing.
3. If no hospital is rejected, stop. Finalize the engagements into the stable matching. Otherwise, rejected hospitals cross the name of the doctor who rejected them off their list and then propose to the favorite among those remaining.
4. Return to Step 2.

List all of your observations about the algorithm here (think: loop invariants):

-
-
-

Claim 1. The Gale-Shapley algorithm returns a stable matching.

Proof.

Claim 2. The Gale-Shapley algorithm terminates in time $O(n^2)$.

Proof.

Claim 3. The hospital-proposing algorithm returns the hospital-optimal stable matching. (Each hospital is matched to their best eligible partner.)

Proof. We call a partner d *eligible* for h if there exists *some* stable matching M' in which (d, h) are partners.

Claim 4. The hospital-proposing algorithm returns the doctor-pessimal stable matching. (Each doctor is matched to their worst eligible partner.)

Proof.