Lab 01: Stable Matching and the Gale-Shapley algorithm

Theory Practice

Question 1

As a second-year student who has not yet been romantically involved with anyone, your best friend invites you to join a date with two other single individuals in hopes of sparking a potential romantic connection. You eagerly anticipate the event and prepare a list of your preferences, considering various scenarios. Each person involved in the date has their own preference list. Provide a scenario where there can be more than one set of couples that are stable over time

Answer:
According to the the problem statement, there are 4 participants in the date, which are denoted as follows:

- Yourself (You) (A)
- Your best friend (B)
- The first singleton (X)
- The second singleton (Y)

One scenario where there can be more than one set of couples that are stable over time is:
The preference list order of each person is as follows:

- A: [X, Y]
- B: [Y, X]
- X: [B, A]
- Y: [A, B]

In this case, there can be 2 sets of couples that are stable over time:

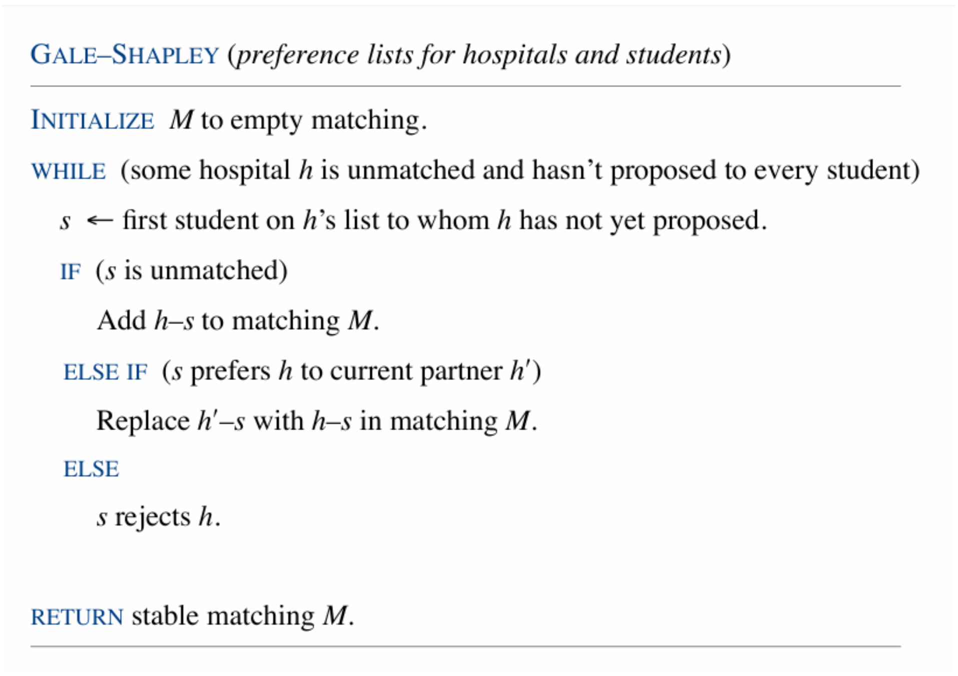
- Pairing 1: A - X and B - Y (You pair with the first singleton and your best friend pairs with the second singleton). In this pairing, A prefers X over Y, so A - Y is not a blocking pair. Next, B prefers Y over X, so B - X is also not a blocking pair. Therefore, this pairing satisfies the property of stable matching.
- Pairing 2: A - Y and B - X (You pair with the second singleton and your best friend pairs with the first singleton). In this pairing, considering the pair A - X, although A prefers X over Y, X prefers B over A, so this is not a blocking pair. Next, considering the pair B - Y, although B prefers Y over X, Y prefers A over B, so this is also not a blocking pair. Therefore, this pairing satisfies the property of stable matching.

Question 2

With the same circumstance as Question 1 - you are still lonely, but you now join a bigger dating event, held by a huge company. There would be many people there, let say N boys and N girls. The company chooses to apply the Gale-Shapley algorithm for matching couples. In the worst case, in which all the boys share the same preference list, how many iterations does it take so that everyone can find their soulmates?

Answer:

- This solution applies the definition of "iterations" as defined by Wikipedia: Gale-Shapley Algorithm on Wikipedia.
- Suppose the Gale-Shapley algorithm is executed with the proposal set being the set of N boys and follows the pseudocode below:



```
GALE–SHAPLEY (preference lists for hospitals and students)

INITIALIZE $M$ to empty matching.
WHILE (some hospital $h$ is unmatched and hasn't proposed to every student)
  $s$ ← first student on $h$'s list to whom $h$ has not yet proposed.
  IF ($s$ is unmatched)
    Add $h$–$s$ to matching $M$.
  ELSE IF ($s$ prefers $h$ to current partner $h'$)
    Replace $h'$–$s$ with $h$–$s$ in matching $M$.
  ELSE
    $s$ rejects $h$.

RETURN stable matching $M$.
```

Figure 1: Gale-Shapley algorithm (preference lists for hospitals and students)

- In the worst case of the Gale-Shapley algorithm, when all the boys have the same preference list, the number of iterations required for everyone to find their soulmates is N. This is because after each iteration, according to the Gale-Shapley algorithm, at least one girl is matched with her "soulmate." With the preference list of the N boys being identical in each iteration, exactly one "new" girl is matched with her "soulmate."
- This occurs because in each iteration, the boys who are not yet matched with their "soulmate" will propose to the same girl. After the "selection" process, the proposed girl will choose to match with the boy who is highest on her priority list. It can be seen that after N iterations like this, both N girls and N boys will be perfectly matched.
- **Conclusion**: The number of iterations required in the worst case for everyone to find their soulmates is N.

Question 3

Consider a bipartite graph $G = (V, E)$ with $X \cup Y = V$ and a set of weights W where $w_{ab}, a, b \in V$. A matching M is stable if there do not exist edges $(x, y), (x', y') \in M$ with $x, x' \in X$ and $y, y' \in Y$ such that $w_{ab'} > w_{ab} \wedge w_{b'a} > w_{ba}$. If M is stable, and all vertices in X share the same set of weights to all vertices in Y, prove that M is unique.

Answer:

I will prove this problem using induction.

Consider the logical expression $P(k)$: "For a set X with k vertices and all vertices in X share the same set of weights to all vertices in Y, there exists exactly one stable matching."

With the constraint that all vertices in X share the same set of weights to all vertices in Y, it can be seen that every vertex in X has a unique preference list. Without loss of generality, we denote this preference list as:

$$\text{order\_list\_X} = \{y_1, y_2, y_3, \ldots, y_k\}$$

Base Case: For $k = 1$, it can be seen that there is exactly one stable matching $M = \{(x_1, y_1)\}$. This matching is unique and also satisfies the condition of stable matching.

Induction Step: Assume that $P(k)$ holds for all $k = 1, \ldots, n-1$. We will prove that $P(n)$ also holds.

Consider the most preferred vertex in order_list_X: y_1.

Consider a stable matching in which x_{y_1} is the vertex matched with y_1. Let $x_{y_1\_best}$ be the vertex that y_1 prefers the most from set X.

If y_1 is not matched with its most preferred vertex $x_{y_1\_best}$ (i.e., $x_{y_1} \neq x_{y_1\_best}$), since all vertices in X share the same order_list_X, $x_{y_1\_best}$ will also prefer y_1 the most. Therefore, when y_1 is matched with x_{y_1} (which is different from $x_{y_1\_best}$), the pair $(x_{y_1\_best}, y_1)$ will become a blocking pair. This contradicts the stability of the stable matching we are considering.

Thus, y_1 must be matched with $x_{y_1\_best}$. When removing these two vertices, the remaining vertices form a problem of size $P(n-1)$, which has been resolved by the assumption.

Therefore, by induction, we can see that the problem $P(n)$ has been proven. Thus, it holds for all n.

Conclusion: The only possible stable matching is:

$$M = \{(x_{y_1\_best}, y_1), (x_{y_2\_best}, y_2), \ldots, (x_{y_n\_best}, y_n)\}$$

Since every stable matching must respect this structure, we conclude that the stable matching is unique.

Question 4

You're hosting your own student chess league where your univerisity team competes against another university. The first stage is in round-robin format. This means each player from one university plays every other player at the opposing university once. To do this, each player has to provide a priority list for which opponent they want to be playing on this given game. As an organizer, you must make sure that every player should be able to be matched with the top choice possible for their first game. Is it possible for there to be a set of players' preferences where you can randomly generate the schedule and still get a stable matching everytime? Justify your answer.

Answer:

There is only one possibility for there to be a set of players' preferences where you can randomly generate the schedule and still get a stable matching every time. When $n = 2$ and the number of players from both universities is equal, the preference lists are as follows:

- $A1 = [B1, B2]$
- $A2 = [B2, B1]$
- $B1 = [A2, A1]$
- $B2 = [A1, A2]$

The players from university 1 are $A1$ and $A2$, while the players from university 2 are $B1$ and $B2$.

In this case, there are two possible matchings:

$$M_1 = \{(A1, B1), (A2, B2)\}$$
$$M_2 = \{(A1, B2), (A2, B1)\}$$

It can be seen that both matchings are stable, similar to the case in Question 1. This confirms that for $n = 2$, it is possible to randomly generate matchings while ensuring stability every time.

Why is this not possible for $n > 2$?

For $n > 2$, we always can construct cases where at least 1 matching will always contain an unstable pair. Because:

- **Preference Cycles and Blocking Pairs**: As the number of players increases, their preferences can create cycles, leading to situations where one player prefers an opponent who prefers another player. This increases the likelihood of blocking pairs, where two players prefer each other over their current matches. For $n > 2$, at least one blocking pair is likely to exist, making it impossible to create a stable matching.