# 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:**

Let's say my friend and I are **A** and **B**, and the two other individuals are **C** and **D**.

Consider the following **ranked preference lists** for potential partners:

| | 1st | 2nd |
|---|---|---|
| A | C | D |
| B | D | C |

Table 1: Ours preference lists

| | 1st | 2nd |
|---|---|---|
| C | B | A |
| D | A | B |

Table 2: Their preference lists

From these preferences, we can identify two stable matchings:

**Matching 1: (A, C) and (B, D)**

- **Checking (A, D):** D prefers A over B, but A still prefers C over D ⇒ (A, D) is **not a blocking pair**.
- **Checking (B, C):** C prefers B over A, but B still prefers D over C ⇒ (B, C) is **not a blocking pair**.
- Since no blocking pairs exist, this is a **stable matching**.

**Matching 2: (A, D) and (B, C)**

- **Checking (A, C):** A prefers C over D, but C still prefers B over A ⇒ (A, C) is **not a blocking pair**.
- **Checking (B, D):** B prefers D over C, but D still prefers A over B ⇒ (B, D) is **not a blocking pair**.
- Again, no blocking pairs exist, making this another **stable matching**.

Thus, in this scenario, there are more than one set of couples that are stable overtime, showing that multiple stable pairings can exist in certain preference structures.

## 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 choose to apply Gale-Shapley algorithm for matching couples. In the worst case, in which all the boys share the same preference list, how many iteration does it take so that everyone can find their soulmates?**

**Answer:**

In the worst case, all boys have the same preference list, leading to the following sequence of events:

### Round 1

- Every boy proposes to his most preferred girl (let's call her Girl 1).
- Girl 1 receives $N$ proposals, selects her most preferred suitor (based on her own preference list), and rejects the rest.
- The rejected boys move on to their second choice in the next round.

### Round 2

- The $N-1$ rejected boys now propose to Girl 2.
- Girl 2 selects her best option and rejects the rest.

### Round $k$ (General Case)

- The $N-k+1$ remaining boys propose to Girl $k$.
- Girl $k$ picks her best option and rejects the rest.

### Final Round ($N$)

- The last remaining boys, having been rejected in all previous rounds, propose to Girl $N$.
- Since this is the last available girl, all proposals are accepted, and the process concludes.

### Total Iterations

Since there are $N$ girls, and each rejected boy must propose to a new girl in each round until all boys are matched. the algorithm will take at most $N$ rounds (iterations) for everyone to be matched. Thus, in the worst case, the algorithm takes $\boldsymbol{N}$ **iterations** to complete.

## 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:**

$$w_{xy'} > w_{xy} \quad \textbf{and} \quad w_{y'x} > w_{y'x'}.$$

**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:**

This follows the same structure as Question 2. Since all $x \in X$ share the same preference list (just like the boys in Q2), the matching process is identical.

Each $x \in X$ initially proposes to their most preferred $y \in Y$, and each $y$ chooses the most preferred $x$ among its suitors. The remaining unmatched $x$ then propose to their second choice in $Y$, and this process repeats until all $x$ or all $y$ are matched.

Since this process is deterministic, the resulting stable matching $M$ is unique.

### More formal proof (using contradiction)

Since all vertices in $X$ share the same set of weights to all vertices in $Y$, this means that each $x \in X$ has the same preference order over $Y$

Let assume that there exist two distinct stable matchings $M$ and $M'$. Since $M \neq M'$, there exist at least two pairs $(x, y)$ and $(x', y') \in M$ that are different from $(x, y')$ and $(x', y) \in M'$.

Since all vertices in $X$ share the same preference order over $Y$, we can assume, without loss of generality, that $x$ and $x'$ prefer $y$ over $y'$.

So in $M$, $y'$ must prefer $x'$ over $x$ for $(x', y')$ to be stable.

But in $M'$, $y'$ must prefer $x$ over $x'$ for $(x, y')$ to be stable.

This leads to a contradiction $\Rightarrow$ our assumption that $M \neq M'$ is a contradiction. We conclude that the stable matching $M$ must be 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:**

It **is possible** for there to be a set of players' preferences to result in stable matchings, even when the schedule is randomly generated. This scenario is the same as Question 1:

| | **1st** | **2nd** |
|---|---|---|
| A | C | D |
| B | D | C |

Table 3: Our university's preference lists

| | **1st** | **2nd** |
|---|---|---|
| C | B | A |
| D | A | B |

Table 4: The other university's preference lists

And from Question 1, we know that the two possible matchings in this scenario are:

- **Matching 1**: (A, C) and (B, D)
- **Matching 2**: (A, D) and (B, C)

Both **matchings** are stable, meaning no two players would prefer each other over their assigned partners. This confirms that under certain preference structures, it is indeed possible to generate a random schedule while still achieving stable matchings every time.