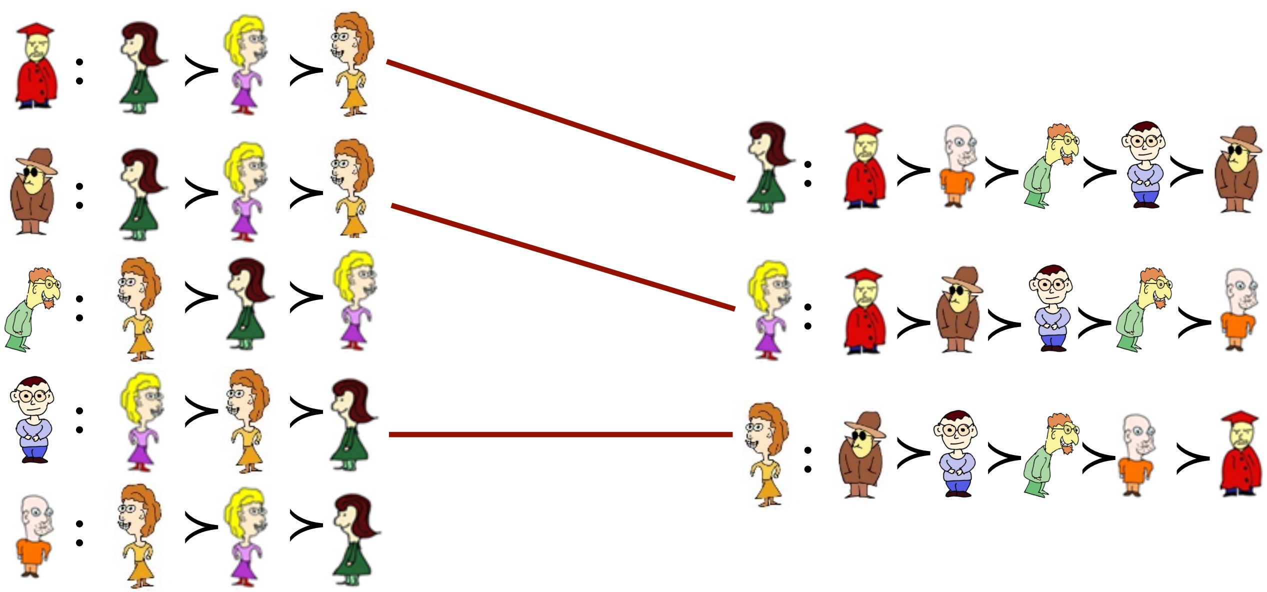
# Computational Social Choice

## Stable Matchings

**Piotr Skowron**
University of Warsaw



# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.

# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.

# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.

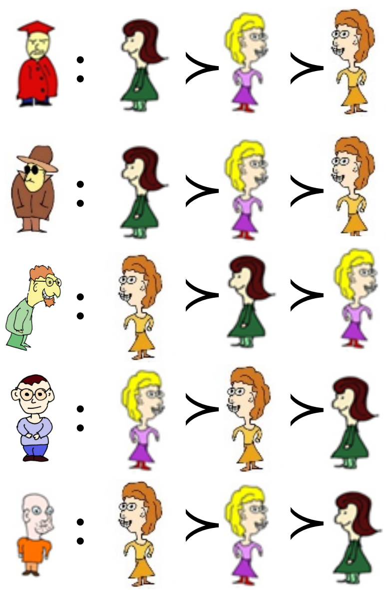
**Goal:** Find a matching between men and women, such that each man is matched with at most one woman, and each woman is matched with at most one man.

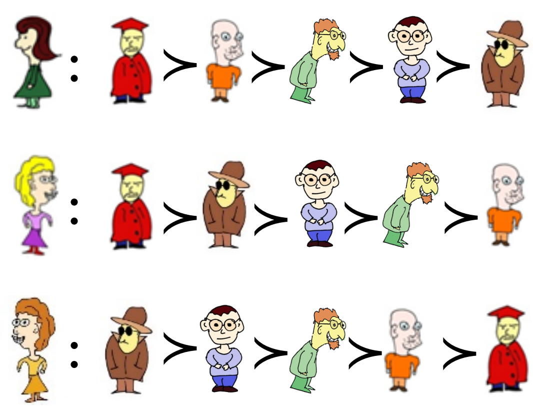
# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.



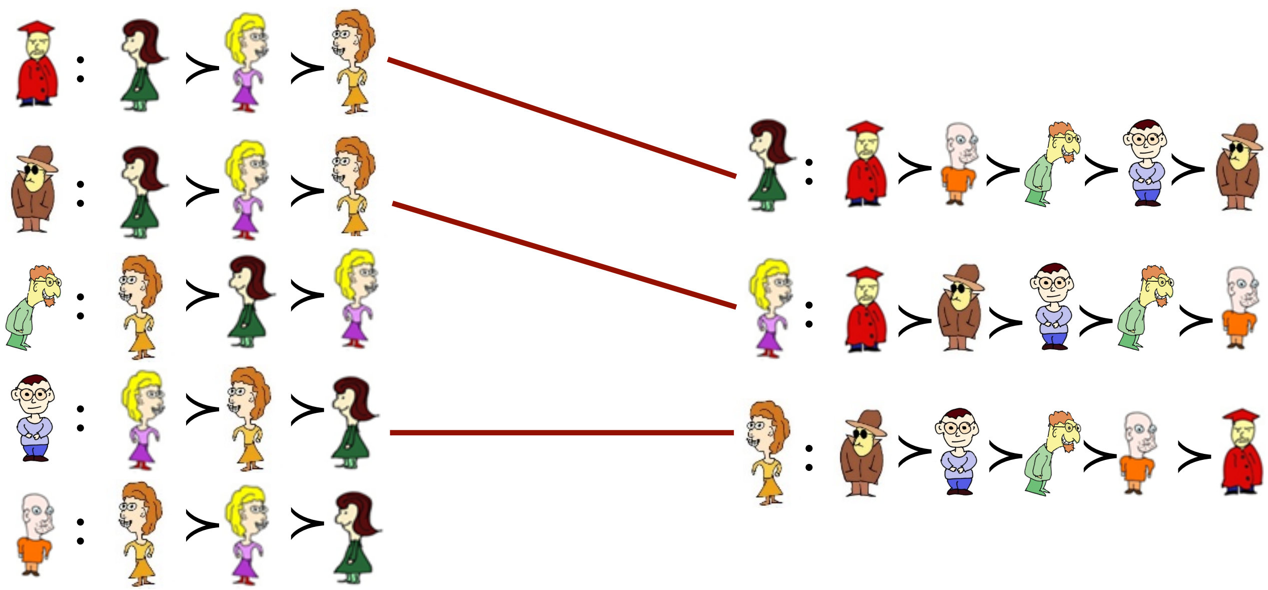


# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.

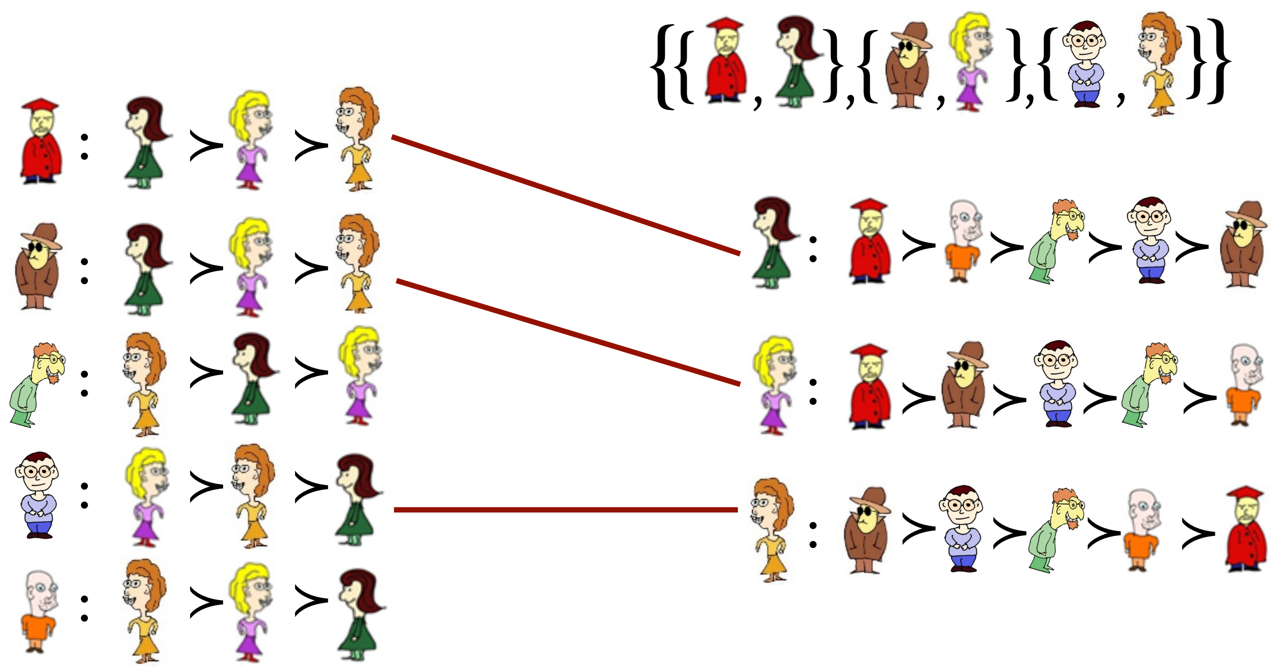


# The model

1. A set of $n$ men $U = \{u_1, u_2, \ldots, u_n\}$.
2. A set of $m$ women $W = \{w_1, w_2, \ldots, w_m\}$.

Each man $u_i$ has a preference relation $\succ_{u_i}$ over the set of women.

Each woman $w_i$ has a preference relation $\succ_{w_i}$ over the set of men.



# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

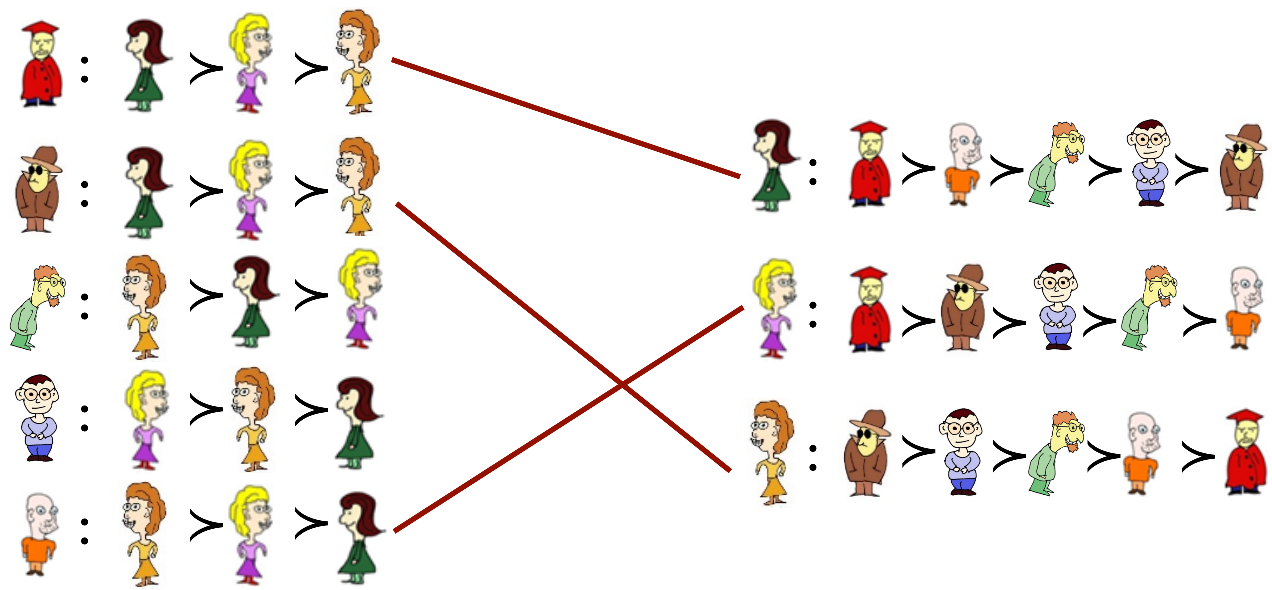
We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u, w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

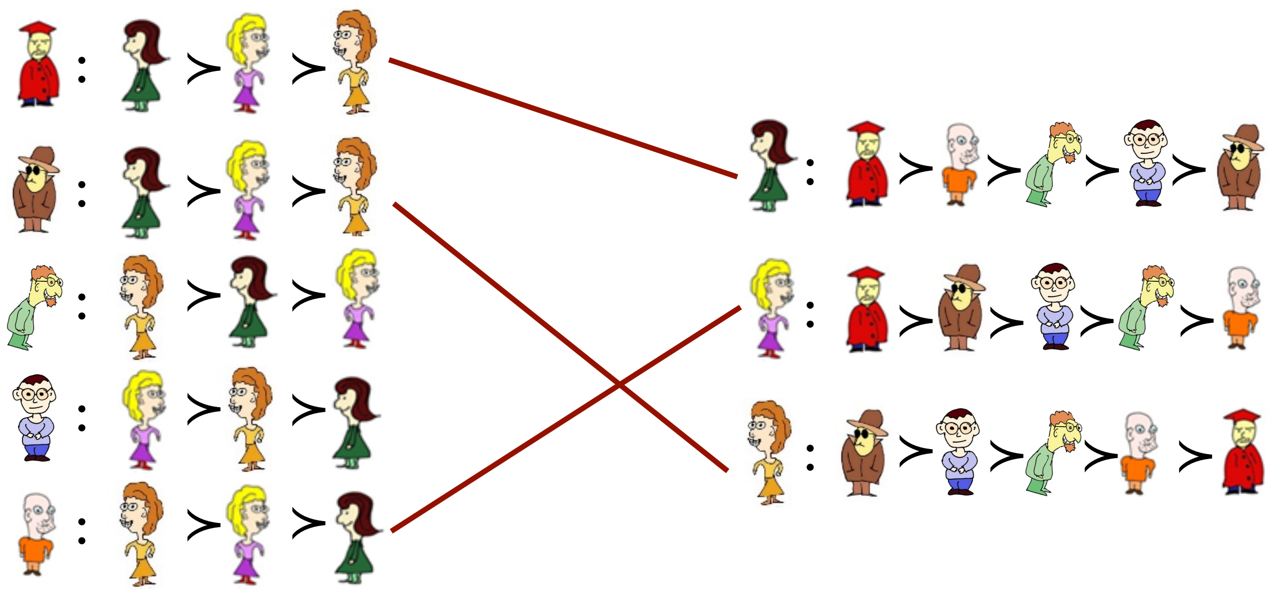


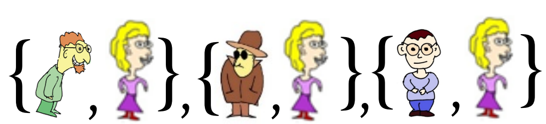
# Stable Matchings

**Stable Matching (CE):** We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.



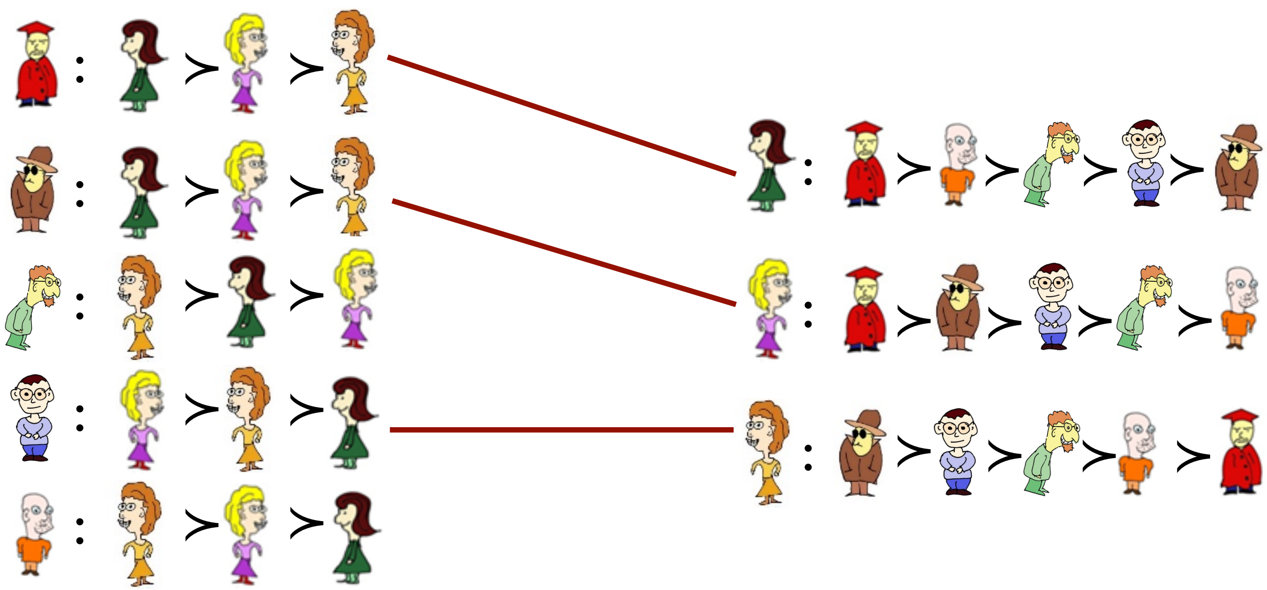
**Blocking pairs:** 

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u, w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

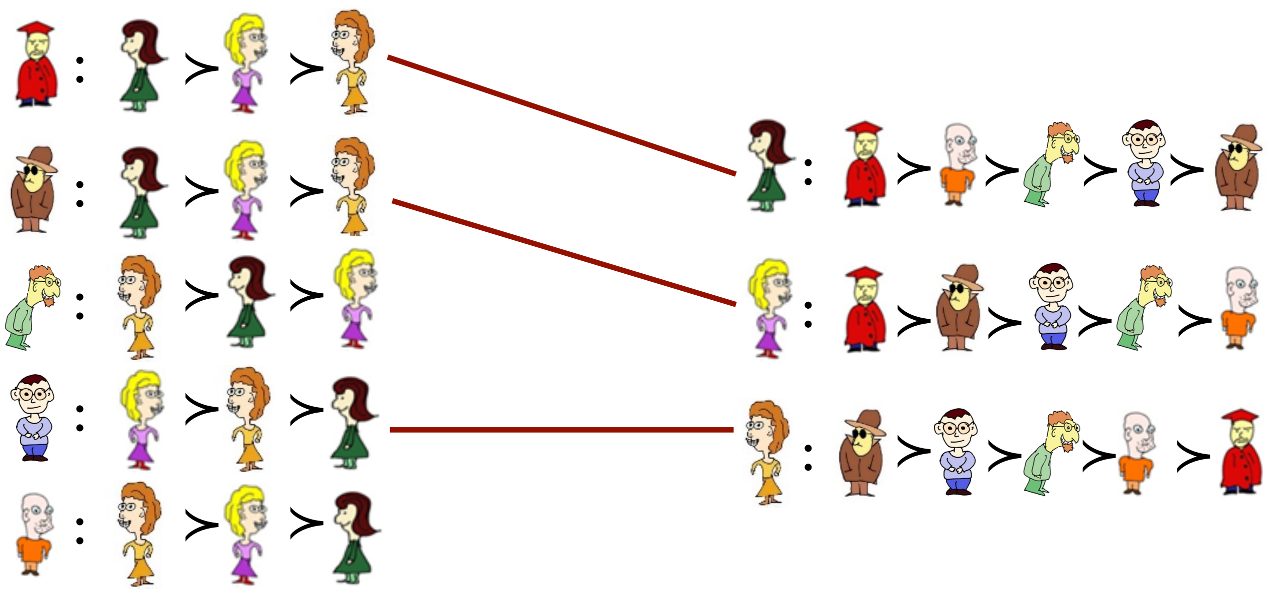


# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.



No blocking pairs. This matching is stable!

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

**Does a stable matching always exists?**

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

**Does a stable matching always exists? If so, how one can find it?**

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u, w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

**Gale-Shapley Algorithm:**

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

**Gale-Shapley Algorithm:**

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.

# Stable Matchings

**Stable Matching (CE)**: We say that a pair of a man and a woman $(u,w)$ blocks a matching $M$ if $u$ and $w$ are not matched and:

1. $u$ is unmatched or prefers $w$ to her partner $M(u)$ in the matching, and
2. $w$ is unmatched or prefers $u$ to her partner $M(w)$ in the matching.

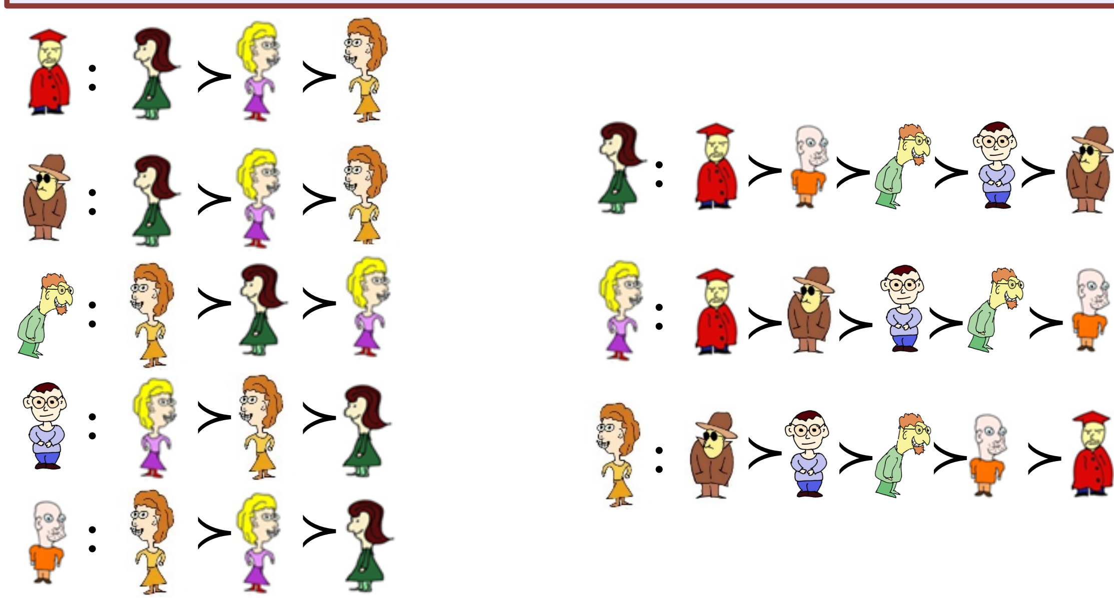
We say that a matching $M$ is stable, if there exists no pair that blocks $M$.

**Gale-Shapley Algorithm:**

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

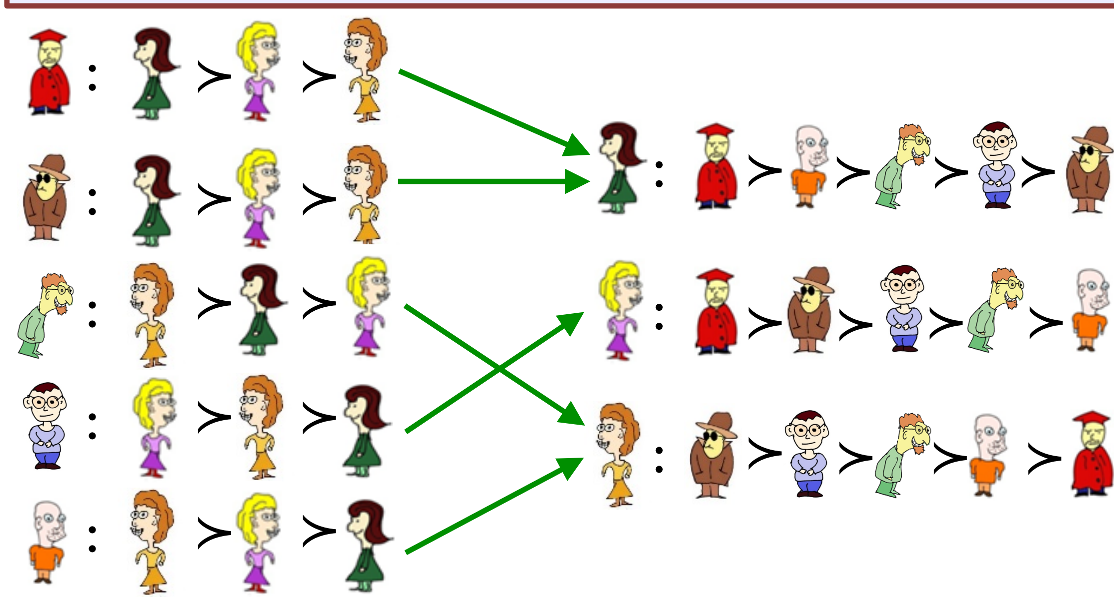
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



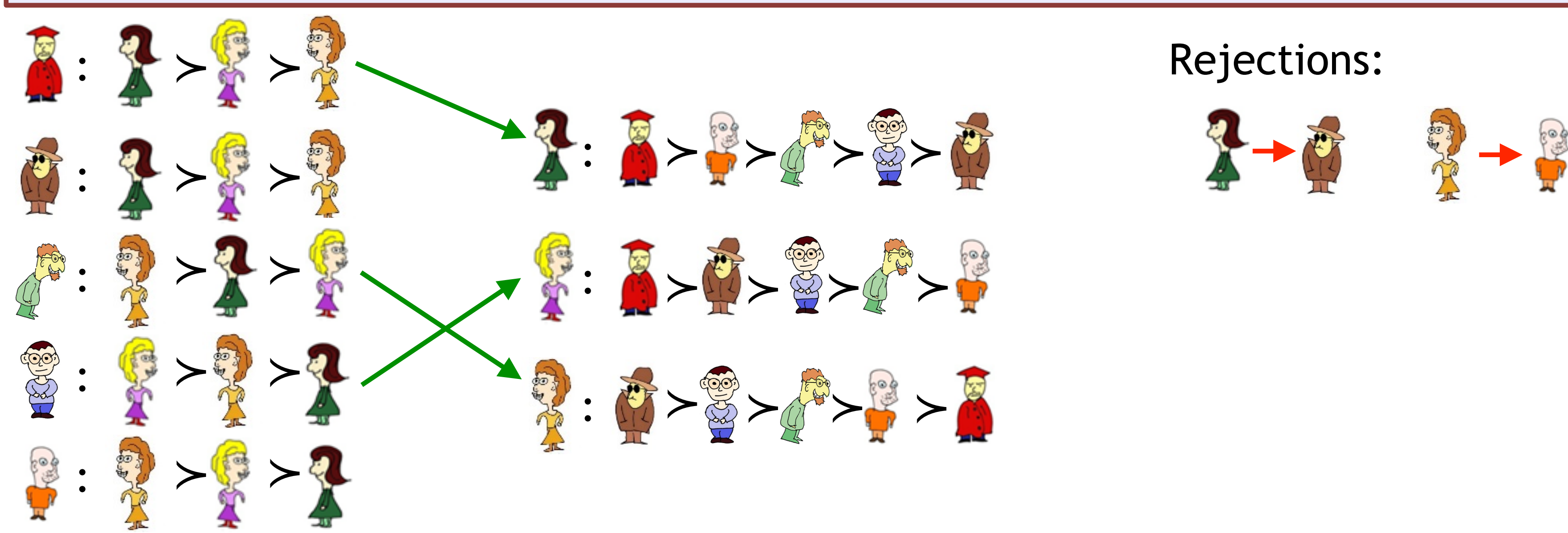
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



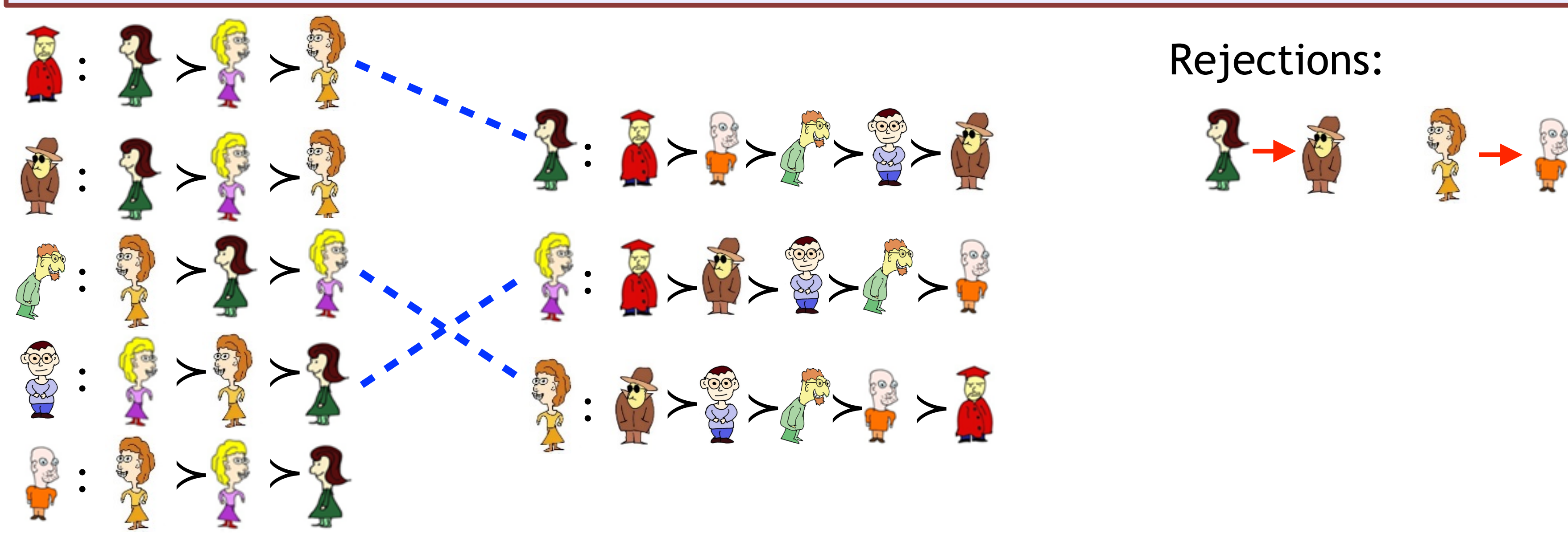
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



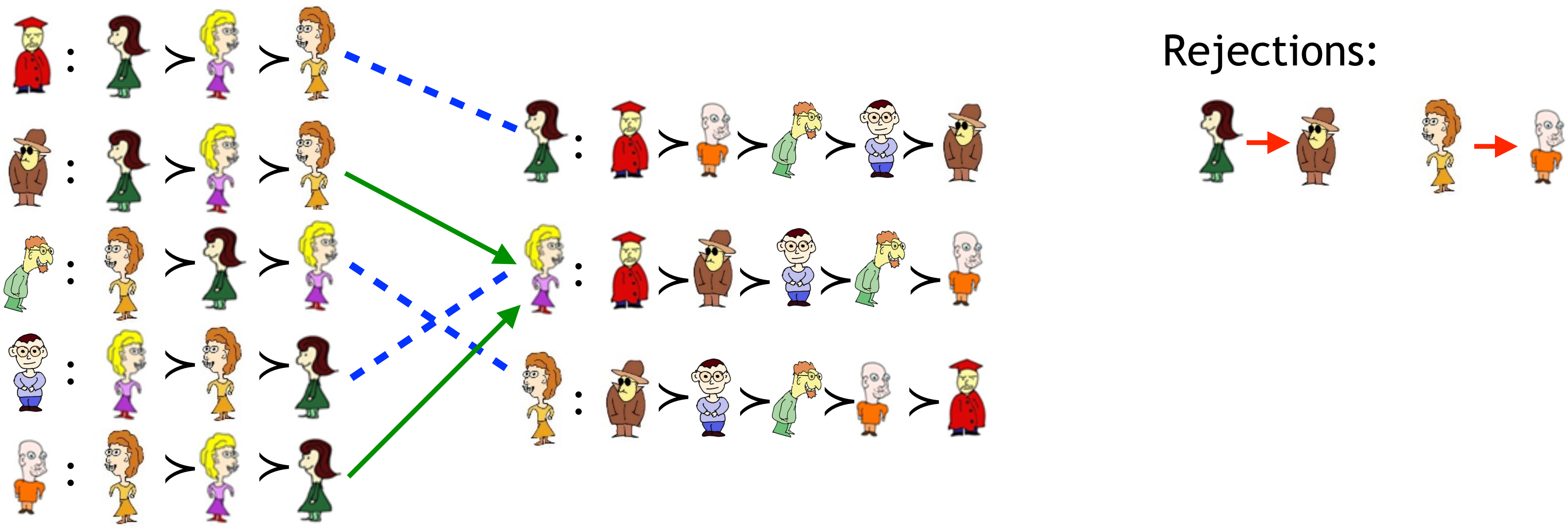
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



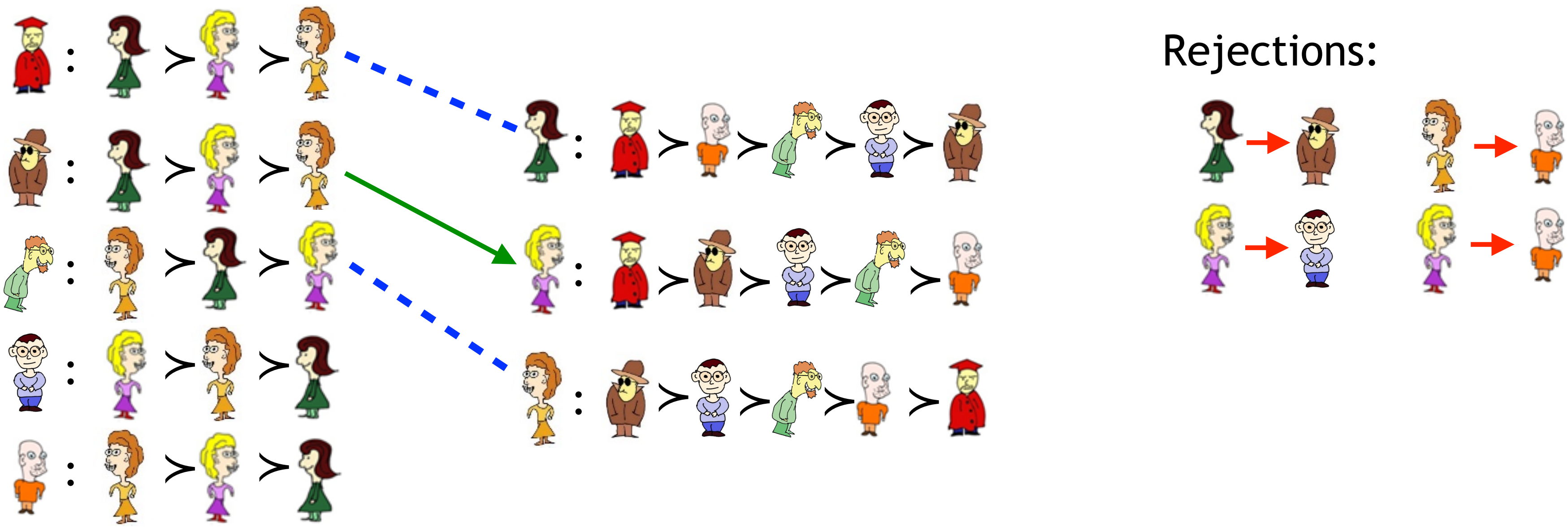
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



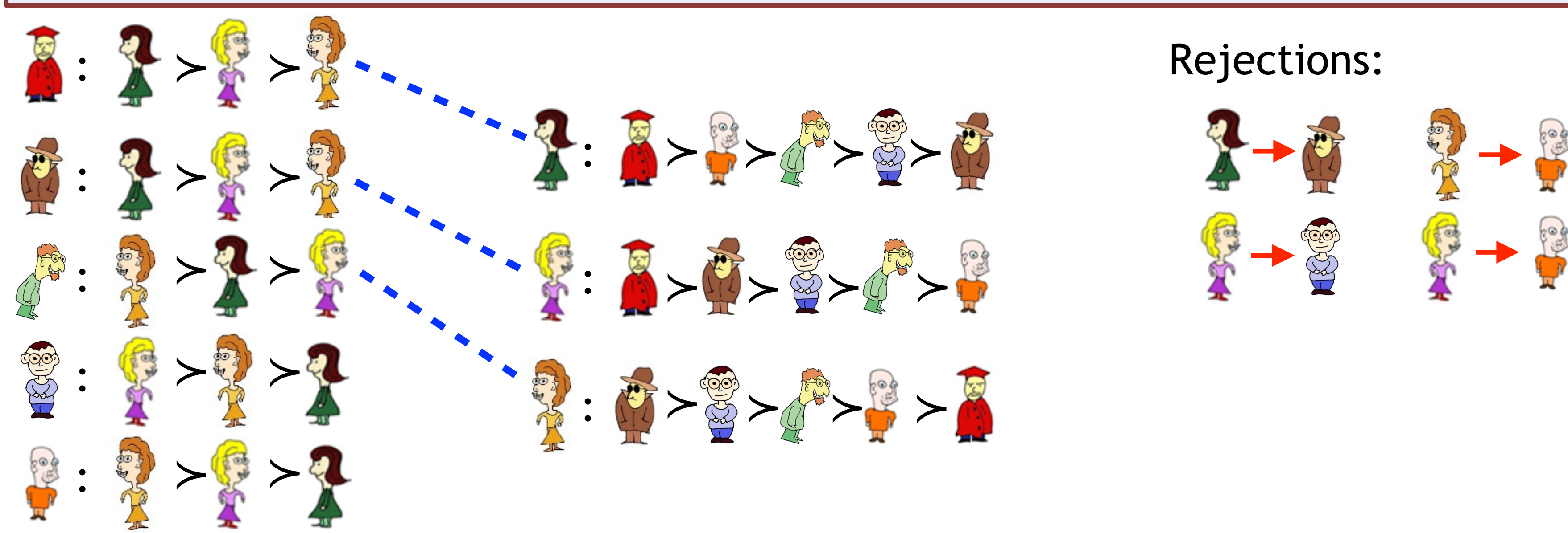
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



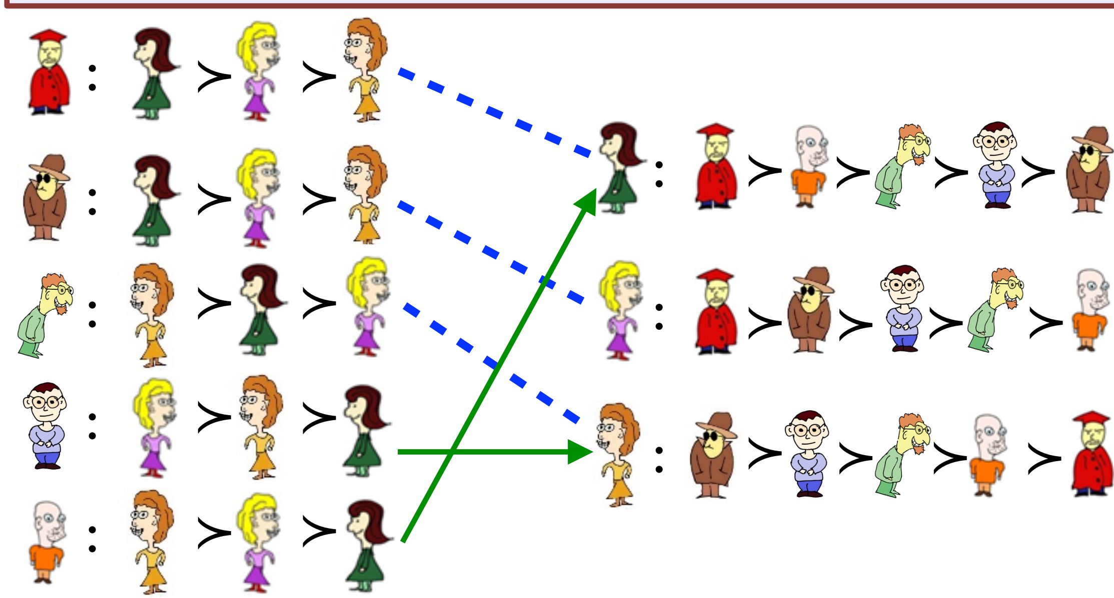
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

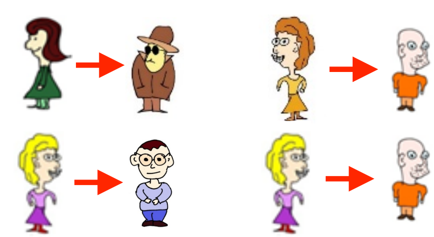


# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

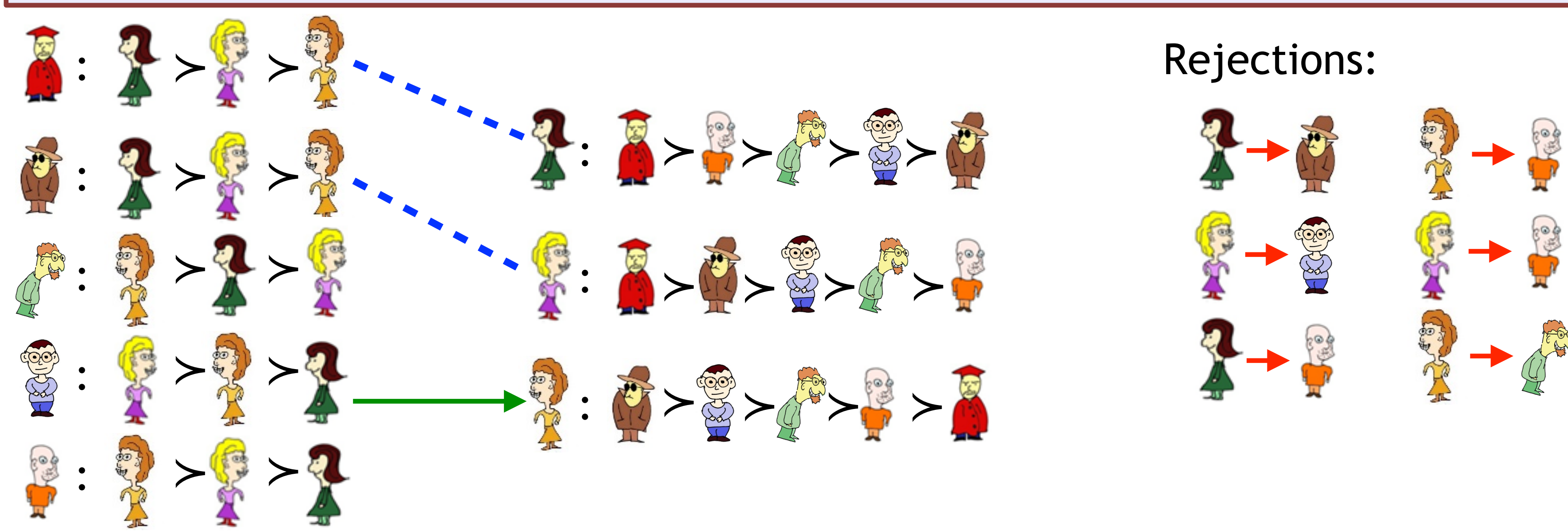


Rejections:



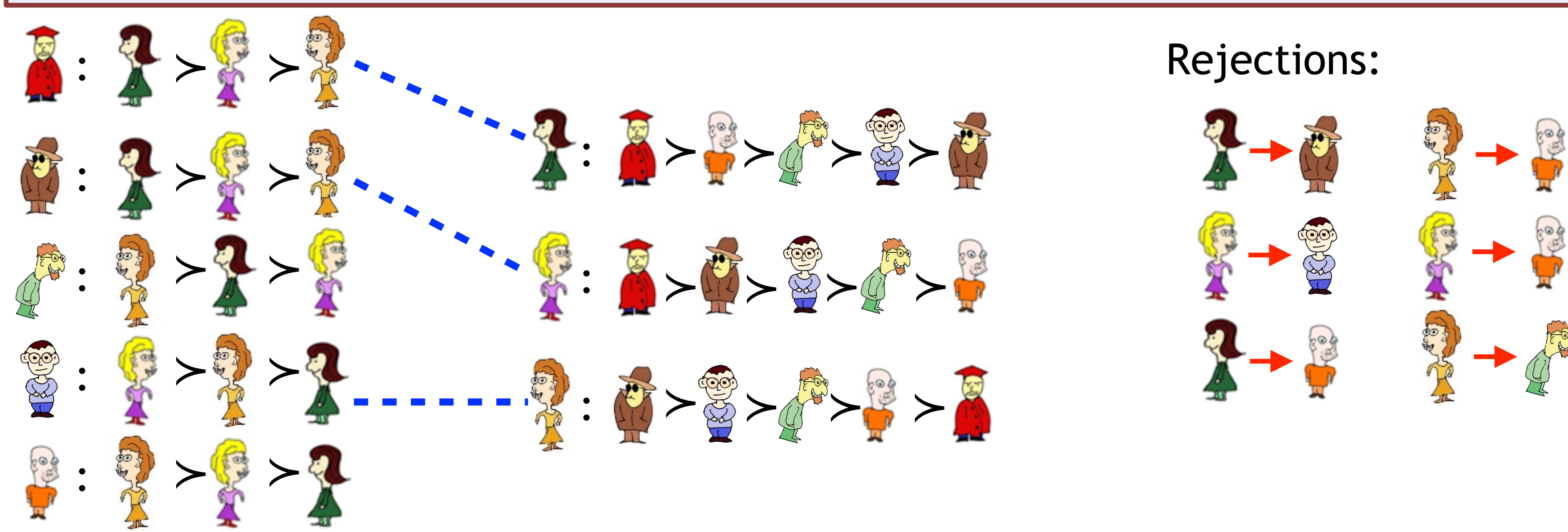
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



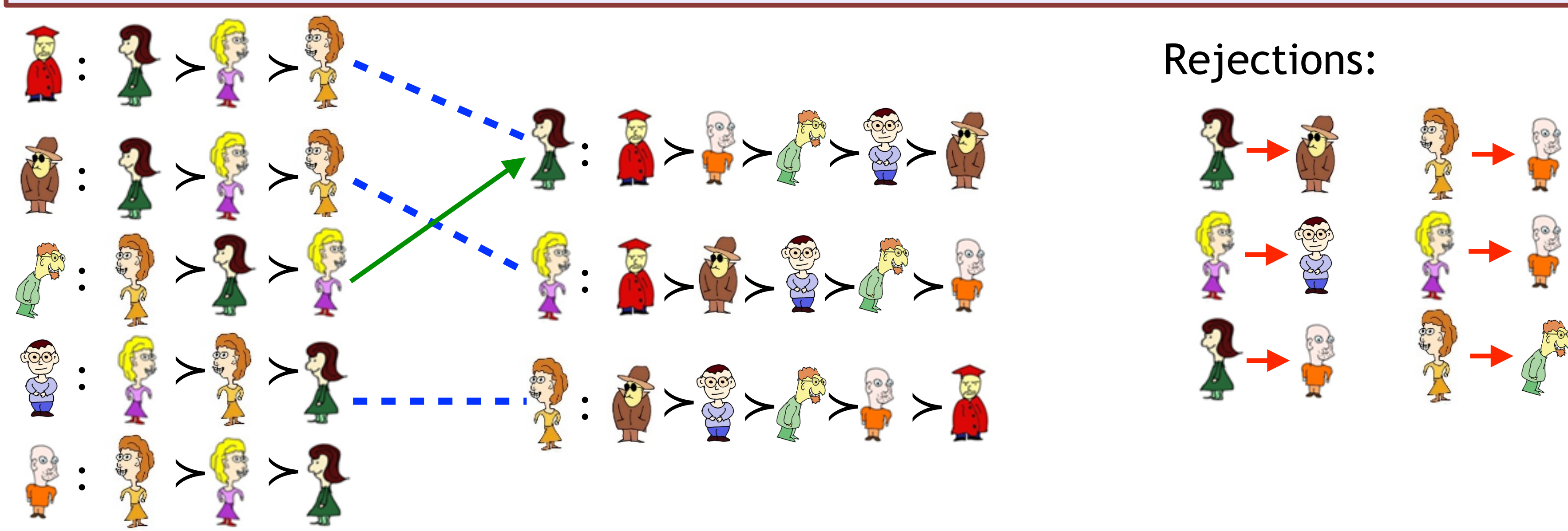
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



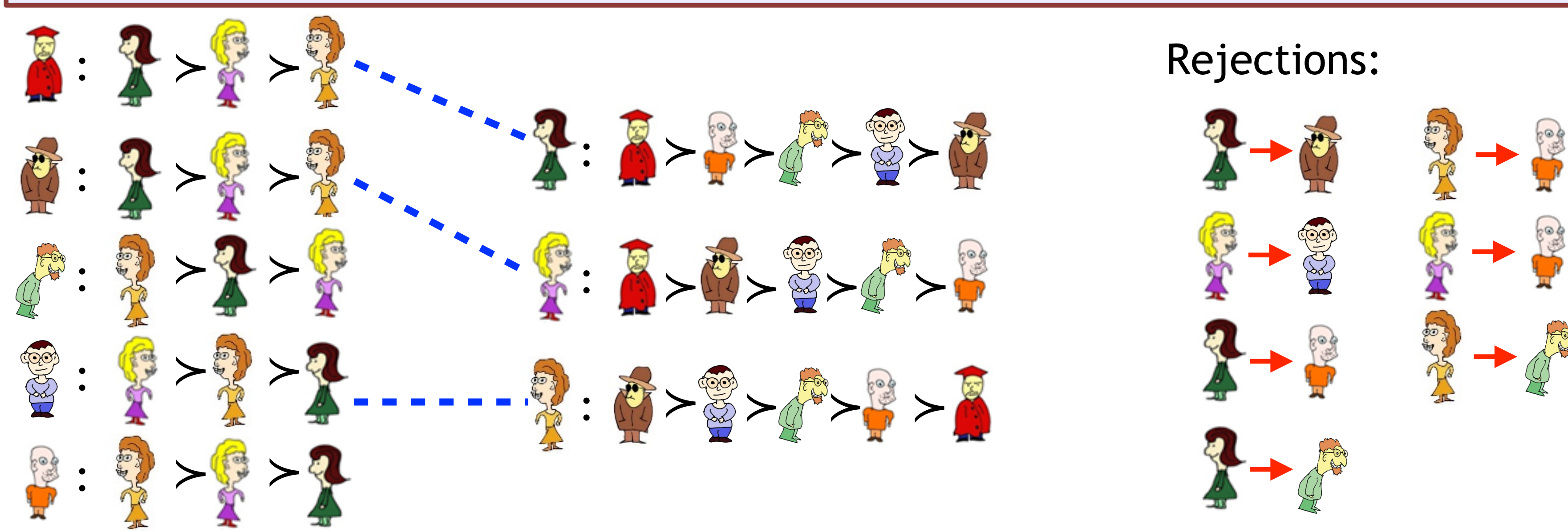
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



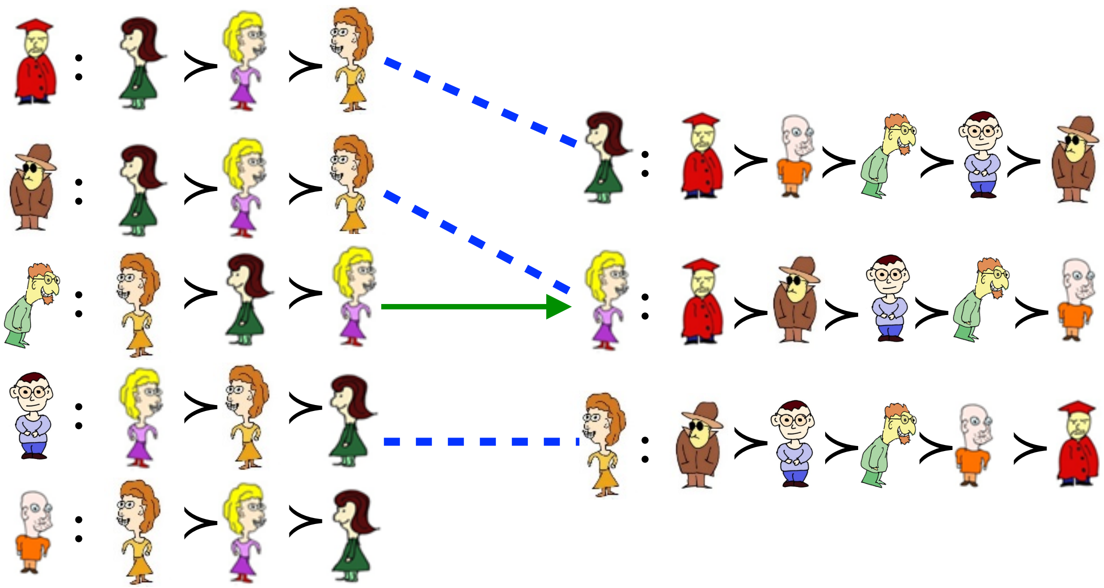
# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

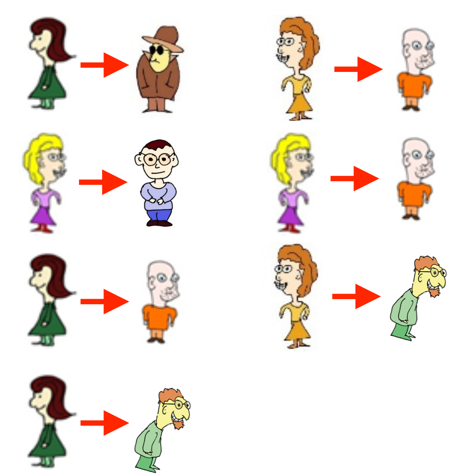


# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



Rejections:

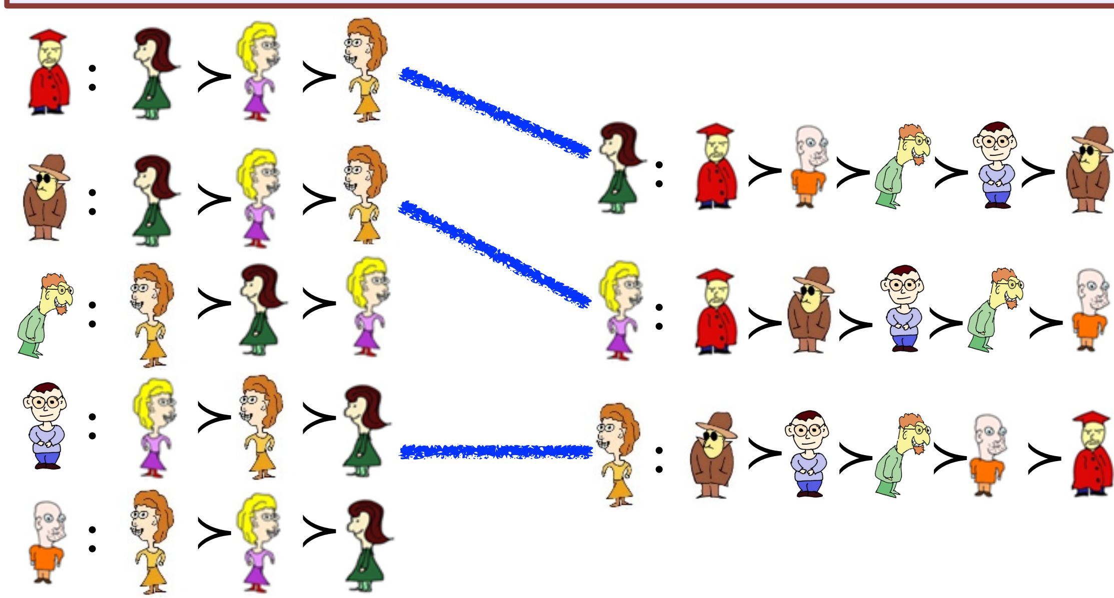


# The Gale-Shapley Algorithm

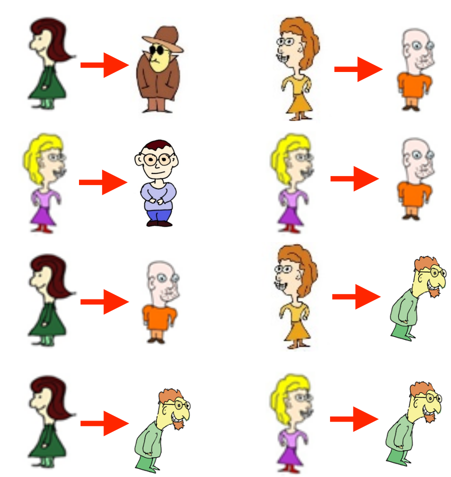
1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.



Rejections:



# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.
Towards a contradiction assume there exists a blocking pair $\{u, w\}$.

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Towards a contradiction assume there exists a blocking pair $\{u, w\}$.

$u$ must have been rejected by $w$ (either because $u$ does not have a partner or because he has a partner who is less preferred).

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Towards a contradiction assume there exists a blocking pair $\{u, w\}$.

$u$ must have been rejected by $w$ (either because $u$ does not have a partner or because he has a partner who is less preferred).

Thus, $w$, when rejecting $u$ was engaged to someone she prefers to $u$.

# The Gale-Shapley Algorithm

1. In the first round each man proposes to his favourite woman. A woman that gets one or multiple proposals picks the man she prefers most, makes a temporary engagement with this man, and rejects all other man.
2. In each subsequent round each unengaged man makes a proposal to his most preferred woman among those who did not reject him. A woman who gets one or multiple proposals picks the one that she prefers most. If she prefers this man to her temporary engaged partner, she breaks this engagement, and makes a temporary engagement with the currently best man among those who proposed. She rejects all proposed man except the one she is engaged to.
3. The process is repeated until every man is either engaged or has been rejected by all women.

**Theorem**: The Gale-Shapley algorithm always produces a stable matching.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Towards a contradiction assume there exists a blocking pair $\{u, w\}$.

$u$ must have been rejected by $w$ (either because $u$ does not have a partner or because he has a partner who is less preferred).

Thus, $w$, when rejecting $u$ was engaged to someone she prefers to $u$.

$w$ breaks an engagement only when getting a better partner.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Assume that in a matching $M'$ there is a man $u_1$ such that:

$$w_2 = M'(u_1) \succ_{u_1} M(u_1) = w_1.$$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Assume that in a matching $M'$ there is a man $u_1$ such that:

$$w_2 = M'(u_1) \succ_{u_1} M(u_1) = w_1.$$

In the Gale-Shapley algorithm $w_2$ must have rejected $u_1$ in favour of $u_2$, who proposed to $w_2$:

$$u_2 \succ_{w_2} u_1.$$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Assume that in a matching $M'$ there is a man $u_1$ such that:

$$w_2 = M'(u_1) \succ_{u_1} M(u_1) = w_1.$$

In the Gale-Shapley algorithm $w_2$ must have rejected $u_1$ in favour of $u_2$, who proposed to $w_2$:

$$u_2 \succ_{w_2} u_1.$$

Since $M'$ is stable, it must be the case that: $w_3 = M'(u_2) \succ_{u_2} w_2$.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Assume that in a matching $M'$ there is a man $u_1$ such that:

$$w_2 = M'(u_1) \succ_{u_1} M(u_1) = w_1.$$

In the Gale-Shapley algorithm $w_2$ must have rejected $u_1$ in favour of $u_2$, who proposed to $w_2$:

$$u_2 \succ_{w_2} u_1.$$

Since $M'$ is stable, it must be the case that: $w_3 = M'(u_2) \succ_{u_2} w_2$.

Since $u_2$ proposed to $w_2$, he must have been rejected by $w_3$ in favour of $u_3$ who proposed to $w_3$.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Consider a matching $M$ returned by the Gale-Shapley algorithm.

Assume that in a matching $M'$ there is a man $u_1$ such that:

$$w_2 = M'(u_1) \succ_{u_1} M(u_1) = w_1.$$

In the Gale-Shapley algorithm $w_2$ must have rejected $u_1$ in favour of $u_2$, who proposed to $w_2$:

$$u_2 \succ_{w_2} u_1.$$

Since $M'$ is stable, it must be the case that: $w_3 = M'(u_2) \succ_{u_2} w_2$.

Since $u_2$ proposed to $w_2$, he must have been rejected by $w_3$ in favour of $u_3$ who proposed to $w_3$.

Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$

$u_1$ $w_1$

$u_2$ $w_2$

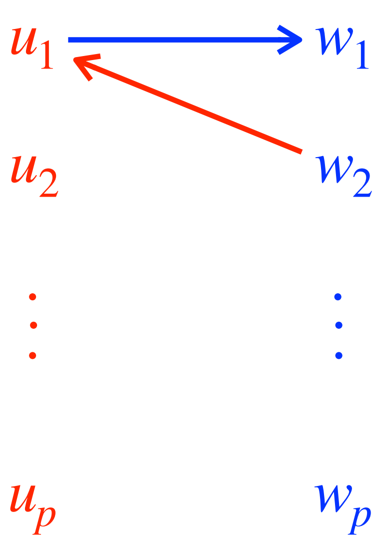
$\vdots$ $\vdots$

$u_p$ $w_p$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$



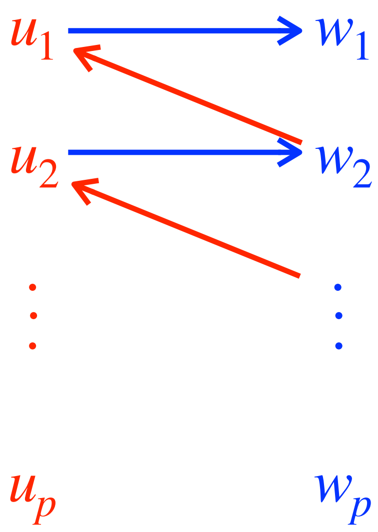

$u_1$ proposed to $w_1$

$w_2$ rejected $u_1$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$



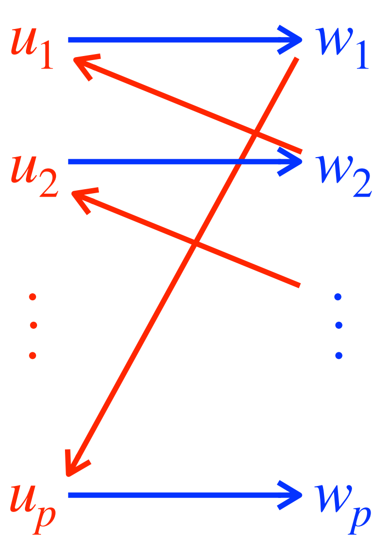

$u_1$ proposed to $w_1$ $w_2$ rejected $u_1$

$u_2$ proposed to $w_2$ $w_3$ rejected $u_2$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$

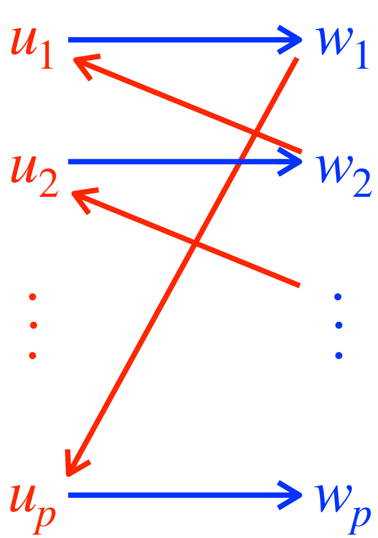



| | |
|---|---|
| $u_1$ proposed to $w_1$ | $w_2$ rejected $u_1$ |
| $u_2$ proposed to $w_2$ | $w_3$ rejected $u_2$ |
| … | … |
| $u_p$ proposed to $w_p$ | $w_1$ rejected $u_p$ |

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$




$u_1$ proposed to $w_1$ | $w_2$ rejected $u_1$

$u_2$ proposed to $w_2$ | $w_3$ rejected $u_2$

… | …

$u_p$ proposed to $w_p$ | $w_1$ rejected $u_p$

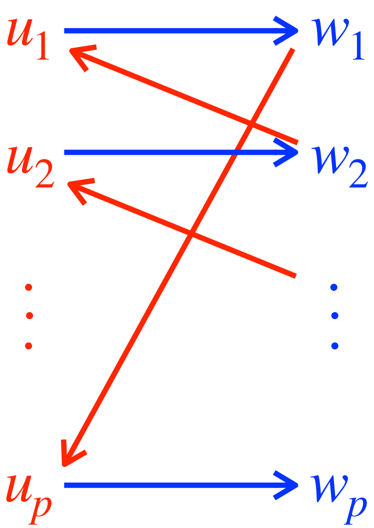
$u_1$ proposed to $w_1$ after he was rejected by $w_2$

$w_2$ rejected $u_1$ after $u_2$ proposed to her.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$




$u_1$ proposed to $w_1$ — $w_2$ rejected $u_1$

$u_2$ proposed to $w_2$ — $w_3$ rejected $u_2$

… — …

$u_p$ proposed to $w_p$ — $w_1$ rejected $u_p$

$u_1$ proposed to $w_1$ after he was rejected by $w_2$

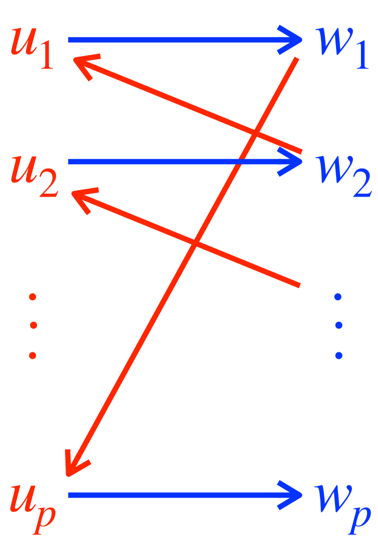
$w_2$ rejected $u_1$ after $u_2$ proposed to her. Thus,

$u_1$ proposed to $w_1$ after $u_2$ proposed to $w_2$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$




| | |
|---|---|
| $u_1$ proposed to $w_1$ | $w_2$ rejected $u_1$ |
| $u_2$ proposed to $w_2$ | $w_3$ rejected $u_2$ |
| ... | ... |
| $u_p$ proposed to $w_p$ | $w_1$ rejected $u_p$ |

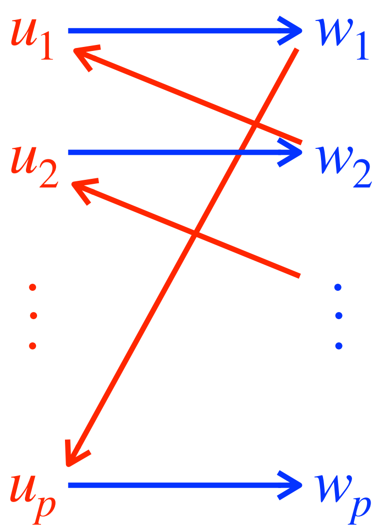
$u_1$ proposed to $w_1$ after he was rejected by $w_2$
$w_2$ rejected $u_1$ after $u_2$ proposed to her. Thus,
$u_1$ proposed to $w_1$ after $u_2$ proposed to $w_2$

similarly, $u_2$ proposed to $w_2$ after $u_3$ proposed to $w_3$

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

Proof: Repeating the reasoning, we find sequences $u_1, u_2, \ldots, u_p$ and $w_1, w_2, \ldots, w_p$




$u_1$ proposed to $w_1$ — $w_2$ rejected $u_1$

$u_2$ proposed to $w_2$ — $w_3$ rejected $u_2$

... — ...

$u_p$ proposed to $w_p$ — $w_1$ rejected $u_p$

$u_1$ proposed to $w_1$ after he was rejected by $w_2$
$w_2$ rejected $u_1$ after $u_2$ proposed to her. Thus,
$u_1$ proposed to $w_1$ after $u_2$ proposed to $w_2$

similarly, $u_2$ proposed to $w_2$ after $u_3$ proposed to $w_3$

And so on, until we get a contradiction.

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

**How to get a woman optimal stable matching?**

# Men-optimal and Women-optimal Stable Matchings

**Theorem**: In any stable matching no man can get a better partner than the one that he gets in the matching returned by the Gale-Shapley algorithm.

**How to get a woman optimal stable matching?**

**Women shall propose instead of men!**

# Applications of  Stable Matchings

1. Matching students to schools.

2. Matching residents to hospitals.

3. Assigning users to servers in a large distributed Internet service.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3
6 proposes to 5

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3
6 proposes to 5 (but gets rejected)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4
5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3
6 proposes to 5 (but gets rejected)
6 proposes to 1

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

1 proposes to 4
6 proposes to 2
3 proposes to 4 (but gets rejected)
3 proposes to 5
2 proposes to 4

5 proposes to 4 (4 rejects 1)
1 proposes to 6 (6 rejects 2)
2 proposes to 3
6 proposes to 5 (but gets rejected)
6 proposes to 1

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 6 | 2 | 5 | 3 |
| 2 | 6 | 3 | 5 | 1 | 4 |
| 3 | 4 | 5 | 1 | 6 | 2 |
| 4 | 2 | 6 | 5 | 1 | 3 |
| 5 | 4 | 2 | 3 | 6 | 1 |
| 6 | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

After this phase one person can be rejected by everyone.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

After this phase one person can be rejected by everyone.

(But then every other person holds a proposal. Why?)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 6 | 2 | 5 | 3 |
| 2 | 6 | 3 | 5 | 1 | 4 |
| 3 | 4 | 5 | 1 | 6 | 2 |
| 4 | 2 | 6 | 5 | 1 | 3 |
| 5 | 4 | 2 | 3 | 6 | 1 |
| 6 | 5 | 1 | 4 | 2 | 3 |

**First phase:** proposals like in a Gale-Shapley algorithm. However, every agent holds keeps her own proposal and the proposal she receives.

After this phase one person can be rejected by everyone.

(But then every other person holds a proposal. Why?)

**Because every person rejected her, and can only improve in the course Of the algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

For the sake of contradiction let $(x, y)$ be the first pair rejected that belongs to some stable matching $M$; $y$ rejected $x$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

For the sake of contradiction let $(x, y)$ be the first pair rejected that belongs to some stable matching $M$; $y$ must have received a better proposal, from $z$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

For the sake of contradiction let $(x, y)$ be the first pair rejected that belongs to some stable matching $M$; $y$ must have received a better proposal, from $z$. In $M$, $z$ must prefer her partner $w$ to $y$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

For the sake of contradiction let $(x, y)$ be the first pair rejected that belongs to some stable matching $M$; $y$ must have received a better proposal, from $z$. In $M$, $z$ must prefer her partner $w$ to $y$. Before $z$ proposed to $y$ she proposed to $w$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 6 | 2 | 5 | 3 |
| 2 | 6 | 3 | 5 | 1 | 4 |
| 3 | 4 | 5 | 1 | 6 | 2 |
| 4 | 2 | 6 | 5 | 1 | 3 |
| 5 | 4 | 2 | 3 | 6 | 1 |
| 6 | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

For the sake of contradiction let $(x, y)$ be the first pair rejected that belongs to some stable matching $M$; $y$ must have received a better proposal, from $z$. In $M$, $z$ must prefer her partner $w$ to $y$. Before $z$ proposed to $y$ she proposed to $w$. Rejection of $z$ by $w$ preceded rejection of $x$ by $y$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

Indeed, $x$ was rejected by anyone who she prefers to $y$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

Indeed, $x$ was rejected by anyone who she prefers to $y$.
If $y$ would have a worse partner than $x$, say $z$, then $(x, y)$ would be blocking.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | 4 | 6 | 2 | 5 | 3 |
| **2** | 6 | 3 | 5 | 1 | 4 |
| **3** | 4 | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | 1 | 3 |
| **5** | 4 | 2 | 3 | 6 | 1 |
| **6** | 5 | 1 | 4 | 2 | 3 |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | | 6 | | | |
| **2** | | 3 | 5 | 1 | 4 |
| **3** | | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | | |
| **5** | 4 | 2 | 3 | | |
| **6** | | 1 | | | |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

**We can truncate preference lists!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | | 6 | | | |
| **2** | | 3 | 5 | 1 | 4 |
| **3** | | 5 | 1 | 6 | 2 |
| **4** | 2 | 6 | 5 | | |
| **5** | 4 | 2 | 3 | | |
| **6** | | 1 | | | |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

**We can truncate preference lists!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** In a stable matching no person can be matched to anyone who she rejected.

**Lemma:** If at any stage $x$ proposed to $y$ then:

1. $x$ cannot have a better partner than $y$, and
2. $y$ cannot have a worse partner than $x$.

**We can truncate preference lists!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

**How to find it?**

$q_i$: the second person in someone's (namely $a_i$'s) list.

$a_{i+1}$: the last person in $q_i$'s list. (so $q_i$ is first in $a_{i+1}$'s list)

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**All-or-nothing cycle:** a sequence $(a_1, a_2, \ldots, a_r)$ such that:

- the second person in $a_i$'s list is the first person in $a_{i+1}$'s list,
- the second person in $a_r$'s list is the first person in $a_1$'s list.

Let $b_i$ denote the first person in $a_i$'s list.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

If $a_i$ is matched to $b_i$ that she considers best, then $a_i$ is worst for $b_i$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

If $a_i$ is matched to $b_i$ that she considers best, then $a_i$ is worst for $b_i$.
If $a_{i+1}$ is not matched to $b_{i+1}$ that she considers best, then she would get a candidate that is worse than $b_i$ (her second choice, matched to $a_i$).

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

If $a_i$ is matched to $b_i$ that she considers best, then $a_i$ is worst for $b_i$.
If $a_{i+1}$ is not matched to $b_{i+1}$ that she considers best, then she would get a candidate that is worse than $b_i$ (her second choice, matched to $a_i$).
Thus, $a_{i+1}$ and $b_i$ would be a blocking pair.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate. Either $x = a_i$ or $x = b_i$ (other candidates have the same partners).

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.
A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate. Either $x = a_i$ or $x = b_i$ (other candidates have the same partners). The case $x = b_i$ is not possible, since $a_i$ gets a better candidate.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$. Since $x$ gets her second choice we know $y$ was removed from the list of $x$ or $y = b_i$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$. Since $x$ gets her second choice we know $y$ was removed from the list of $x$ or $y = b_i$. The latter is not possible since $b_i$ prefers her new partner.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$. Since $x$ gets her second choice we know $y$ was removed from the list of $x$.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.

A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$. Since $x$ gets her second choice we know $y$ was removed from the list of $x$. If $x$ proposed to $y$ and got rejected then $x$ preferred $y$ to her matched candidate.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

Let $A = \{a_1, a_2, \ldots a_r\}$ and $B = \{b_1, b_2, \ldots b_r\}$.
A new blocking pair $(x, y)$ such that $x$ preferred her partner to $y$, but received a worse candidate; $x = a_i$. Since $x$ gets her second choice we know $y$ was removed from the list of $x$. If $x$ proposed to $y$ and got rejected then $x$ preferred $y$ to her matched candidate. So $y$ removed herself from the list of $x$ (got better).

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**The cycle can be eliminated!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | 5 | | | 2 |
| 4 | 2 | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**The cycle can be eliminated!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | | 6 | | | |
| **2** | | 3 | 5 | | 4 |
| **3** | | | | | 2 |
| **4** | | | 5 | | |
| **5** | 4 | 2 | 3 | | |
| **6** | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**The cycle can be eliminated!**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**If some list is empty no stable matching exists.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**We continue the propose-reject algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | 5 | | 4 |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

We continue the propose-reject algorithm.

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | | 6 | | | |
| **2** | | 3 | | | |
| **3** | | | | | 2 |
| **4** | | | 5 | | |
| **5** | 4 | 2 | 3 | | |
| **6** | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**We continue the propose-reject algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | | | |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**We continue the propose-reject algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | | | |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | 2 | 3 | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**We continue the propose-reject algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| **1** | | 6 | | | |
| **2** | | 3 | | | |
| **3** | | | | | 2 |
| **4** | | | 5 | | |
| **5** | 4 | | | | |
| **6** | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**We continue the propose-reject algorithm.**

# Stable Roommate: non-bipartite graphs

| | | | | | |
|---|---|---|---|---|---|
| 1 | | 6 | | | |
| 2 | | 3 | | | |
| 3 | | | | | 2 |
| 4 | | | 5 | | |
| 5 | 4 | | | | |
| 6 | | 1 | | | |

**Lemma:** If some $a_i$ gets her first choice in a stable-matching, then so all others on the cycle.

**Lemma:** If in the stable matching each person in the cycle gets her first choice, then after shifting the cycle, the matching would still be stable.

**If each list contains a single element we have a stable matching.**

# Literature

https://en.wikipedia.org/wiki/Stable_marriage_problem

Robert W Irving . An efficient algorithm for the "stable roommates" problem. Journal of Algorithms Volume 6, Issue 4, December 1985, Pages 577-595.

**Further reading:**

Robert W. Irving and Paul Leather. The Complexity of Counting Stable Marriages. SIAM Journal on Computing, Volume 15, Issue 3 (1986).

Robert W. Irving, Paul Leather, and Dan Gusfield. An efficient algorithm for the "optimal" stable marriage. Journal of the ACM, Volume 34, Issue 3, pages 532–543 (1987).