

# Tópicos de Análise de Algoritmos

## Teoria dos Jogos Algorítmica

### Casamentos estáveis

Sec 1.1 do KT. Por curiosidade, leia também o verbete sobre *Stable marriage problem* da wiki.

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

$H$  - conjunto de homens

$M$  - conjunto de mulheres

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

$H$  - conjunto de homens

$M$  - conjunto de mulheres

Cada  $h$  tem ordem estrita de preferência sobre  $M$ .

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

$H$  - conjunto de homens

$M$  - conjunto de mulheres

Cada  $h$  tem ordem estrita de preferência sobre  $M$ .

Cada  $m$  tem ordem estrita de preferência sobre  $H$ .

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

$H$  - conjunto de homens

$M$  - conjunto de mulheres

Cada  $h$  tem ordem estrita de preferência sobre  $M$ .

Cada  $m$  tem ordem estrita de preferência sobre  $H$ .

Adicione homem/mulher fictício para  
representar a possibilidade de ficar solteiro/solteira.

Assim  $|H| = |M|$ .

# Casamentos

Atribuição de médicos a programas de residência,  
usuários a servidores em grandes redes distribuídas de Internet,  
estudantes a universidades, etc.

$H$  - conjunto de homens

$M$  - conjunto de mulheres

Cada **homem** tem ordem estrita de preferência sobre  $M$ .

Cada **mulher** tem ordem estrita de preferência sobre  $H$ .

Adicione homem/mulher fictício para  
representar a possibilidade de ficar solteiro/solteira.

Assim  $|H| = |M|$ .

**Emparelhamento** de  $H$  em  $M$ : alocação de homens a mulheres.

# Casamentos estáveis e pares bloqueadores

$H$  - conjunto de homens       $M$  - conjunto de mulheres

Cada **homem** tem ordem estrita de preferência sobre  $M$ .

Cada **mulher** tem ordem estrita de preferência sobre  $H$ .

**Emparelhamento** de  $H$  em  $M$ : alocação de homens a mulheres.

# Casamentos estáveis e pares bloqueadores

$H$  - conjunto de homens       $M$  - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre  $M$ .

Cada mulher tem ordem estrita de preferência sobre  $H$ .

Emparelhamento de  $H$  em  $M$ : alocação de homens a mulheres.

Um emparelhamento é **instável** se existem homens  $h$  e  $h'$  e mulheres  $m$  e  $m'$  tq  $m$  é emparelhada com  $h$ ,  $m'$  com  $h'$ , mas  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

# Casamentos estáveis e pares bloqueadores

$H$  - conjunto de homens       $M$  - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre  $M$ .

Cada mulher tem ordem estrita de preferência sobre  $H$ .

Emparelhamento de  $H$  em  $M$ : alocação de homens a mulheres.

Um emparelhamento é **instável** se existem homens  $h$  e  $h'$  e mulheres  $m$  e  $m'$  tq  $m$  é emparelhada com  $h$ ,  $m'$  com  $h'$ , mas  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

Neste caso,  $h$  e  $m'$  preferiam se casar um com o outro.

# Casamentos estáveis e pares bloqueadores

$H$  - conjunto de homens       $M$  - conjunto de mulheres

Cada homem tem ordem estrita de preferência sobre  $M$ .

Cada mulher tem ordem estrita de preferência sobre  $H$ .

Emparelhamento de  $H$  em  $M$ : alocação de homens a mulheres.

Um emparelhamento é **instável** se existem homens  $h$  e  $h'$  e mulheres  $m$  e  $m'$  tq  $m$  é emparelhada com  $h$ ,  $m'$  com  $h'$ , mas  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

Neste caso,  $h$  e  $m'$  preferiam se casar um com o outro.

O par  $(h, m')$  é um **par bloqueador**,  
e um emparelhamento é **estável** se não tem par bloqueador.

## Casamentos estáveis: exemplo

Ordens de preferências para  $n = 3$ :

# Casamentos estáveis: exemplo

Ordens de preferências para  $n = 3$ :

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

# Casamentos estáveis: exemplo

Ordens de preferências para  $n = 3$ :

| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

O emparelhamento  $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$  é instável, pois  $(h_1, m_2)$  é um **par bloqueador**.

# Casamentos estáveis: exemplo

Ordens de preferências para  $n = 3$ :

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

O emparelhamento  $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$  é instável, pois  $(h_1, m_2)$  é um par bloqueador.

Já o emparelhamento  $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$  é estável.

# Casamentos estáveis: exemplo

Ordens de preferências para  $n = 3$ :

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

O emparelhamento  $\{(h_1, m_1), (h_2, m_2), (h_3, m_3)\}$  é instável, pois  $(h_1, m_2)$  é um par bloqueador.

Já o emparelhamento  $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$  é estável.

Dadas as listas de preferências de todos, existe emparelhamento estável?

Como encontrar um, se existe?

# Algoritmo da aceitação postergada (Gale e Shapley)

Versão com proposta masculina:

# Algoritmo da aceitação postergada (Gale e Shapley)

Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

# Algoritmo da aceitação postergada (Gale e Shapley)

## Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Para esse, posterga a sua resposta.

# Algoritmo da aceitação postergada (Gale e Shapley)

## Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Para esse, posterga a sua resposta.

## Nova rodada:

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

# Algoritmo da aceitação postergada (Gale e Shapley)

## Versão com proposta masculina:

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Para esse, posterga a sua resposta.

## Nova rodada:

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Eventualmente recusa proposta recebida em rodada anterior.

# Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista. Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

# Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista. Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Repete esse processo,  
sempre progredindo nas listas dos homens.

# Algoritmo com proposta masculina

Cada homem propõe à primeira mulher de sua lista.

Cada mulher que recebeu mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista. Para esse, posterga a sua resposta.

Cada homem que teve sua proposta recusada propõe à próxima mulher de sua lista.

Cada mulher com mais de uma proposta recusa todas, exceto a do homem mais alto em sua lista.

Repete esse processo,  
sempre progredindo nas listas dos homens.

O processo então termina (em não mais que  $n^2$  rodadas).

## Algoritmo no exemplo

| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ |
|---------------|---------------|---------------|
| $m_2$         | $m_1$         | $m_1$         |
| $m_1$         | $m_3$         | $m_2$         |
| $m_3$         | $m_2$         | $m_3$         |

| $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
|---------------|---------------|---------------|
| $h_1$         | $h_3$         | $h_1$         |
| $h_3$         | $h_1$         | $h_3$         |
| $h_2$         | $h_2$         | $h_2$         |

## Algoritmo no exemplo

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

Primeira rodada:

$h_1$  propõe para  $m_2$

$h_2$  propõe para  $m_1$

$h_3$  propõe para  $m_1$

## Algoritmo no exemplo

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

Primeira rodada:

$h_1$  propõe para  $m_2$        $h_2$  propõe para  $m_1$        $h_3$  propõe para  $m_1$   
 $m_1$  rejeita a proposta de  $h_2$ .

## Algoritmo no exemplo

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

### Primeira rodada:

$h_1$  propõe para  $m_2$        $h_2$  propõe para  $m_1$        $h_3$  propõe para  $m_1$   
 $m_1$  rejeita a proposta de  $h_2$ .

### Segunda rodada:

$h_2$  propõe para  $m_3$ , e o algoritmo termina.

## Algoritmo no exemplo

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

### Primeira rodada:

$h_1$  propõe para  $m_2$        $h_2$  propõe para  $m_1$        $h_3$  propõe para  $m_1$   
 $m_1$  rejeita a proposta de  $h_2$ .

### Segunda rodada:

$h_2$  propõe para  $m_3$ , e o algoritmo termina.

**Emparelhamento produzido:**  $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

# Algoritmo no exemplo

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

## Primeira rodada:

$h_1$  propõe para  $m_2$        $h_2$  propõe para  $m_1$        $h_3$  propõe para  $m_1$   
 $m_1$  rejeita a proposta de  $h_2$ .

## Segunda rodada:

$h_2$  propõe para  $m_3$ , e o algoritmo termina.

**Emparelhamento produzido:**  $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

Note que tal emparelhamento é estável!

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Esboço da prova:

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Esboço da prova:

Suponha, por contradição, que o emparelhamento não é estável e sejam  $(h, m)$  e  $(h', m')$  casais tais que  $(h, m')$  é um par bloqueador.

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Esboço da prova:

Suponha, por contradição, que o emparelhamento não é estável e sejam  $(h, m)$  e  $(h', m')$  casais tais que  $(h, m')$  é um par bloqueador.

Ou seja,  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Esboço da prova:

Suponha, por contradição, que o emparelhamento não é estável e sejam  $(h, m)$  e  $(h', m')$  casais tais que  $(h, m')$  é um par bloqueador.

Ou seja,  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

Como  $m' \succ_h m$ , pelo algoritmo,  $h$  propôs para  $m'$  antes de propor a  $m$  e foi rejeitado.

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

Esboço da prova:

Suponha, por contradição, que o emparelhamento não é estável e sejam  $(h, m)$  e  $(h', m')$  casais tais que  $(h, m')$  é um par bloqueador.

Ou seja,  $m' \succ_h m$  e  $h \succ_{m'} h'$ .

Como  $m' \succ_h m$ , pelo algoritmo,  $h$  propôs para  $m'$  antes de propor a  $m$  e foi rejeitado.

Então  $m'$  estava com um homem mais bem colocado que  $h$  na sua lista de preferências, e teria terminado com um homem mais bem colocado que  $h$ , e não com  $h'$  que é tal que  $h \succ_{m'} h'$ . □

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

**Emparelhamento produzido:**  $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

No exemplo, aplicando a versão da **proposta feminina**, obtemos o emparelhamento  $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$ .

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

**Emparelhamento produzido:**  $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

No exemplo, aplicando a versão da **proposta feminina**, obtemos o emparelhamento  $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$ .

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

Diferente do obtido pela proposta masculina!

# Estabilidade do emparelhamento

**Teorema:** O emparelhamento produzido pelo algoritmo da aceitação postergada é estável.

**Emparelhamento produzido:**  $\{(h_1, m_2), (h_2, m_3), (h_3, m_1)\}$

No exemplo, aplicando a versão da **proposta feminina**, obtemos o emparelhamento  $\{(h_1, m_1), (h_2, m_3), (h_3, m_2)\}$ .

|               |               |               |               |               |               |
|---------------|---------------|---------------|---------------|---------------|---------------|
| $\succ_{h_1}$ | $\succ_{h_2}$ | $\succ_{h_3}$ | $\succ_{m_1}$ | $\succ_{m_2}$ | $\succ_{m_3}$ |
| $m_2$         | $m_1$         | $m_1$         | $h_1$         | $h_3$         | $h_1$         |
| $m_1$         | $m_3$         | $m_2$         | $h_3$         | $h_1$         | $h_3$         |
| $m_3$         | $m_2$         | $m_3$         | $h_2$         | $h_2$         | $h_2$         |

Diferente do obtido pela proposta masculina!

Há alguma diferença significativa entre estes emparelhamentos?

# Emparelhamentos ótimos

Emparelhamento  $\nu$  é **ótimo-masculino** se não há emparelhamento estável  $\mu$  tq  $\mu(h) \succ_h \nu(h)$  ou  $\mu(h) = \nu(h)$  para todo  $h$  em  $H$  e  $\mu(j) \succ_j \nu(j)$  para algum  $j$  em  $H$ .

# Emparelhamentos ótimos

Emparelhamento  $\nu$  é **ótimo-masculino** se não há emparelhamento estável  $\mu$  tq  $\mu(h) \succ_h \nu(h)$  ou  $\mu(h) = \nu(h)$  para todo  $h$  em  $H$  e  $\mu(j) \succ_j \nu(j)$  para algum  $j$  em  $H$ .

Definição de emparelhamento **ótimo-feminino** é análoga.

# Emparelhamentos ótimos

Emparelhamento  $\nu$  é **ótimo-masculino** se não há emparelhamento estável  $\mu$  tq  $\mu(h) \succ_h \nu(h)$  ou  $\mu(h) = \nu(h)$  para todo  $h$  em  $H$  e  $\mu(j) \succ_j \nu(j)$  para algum  $j$  em  $H$ .

Definição de emparelhamento **ótimo-feminino** é análoga.

**Teorema:** O algoritmo da aceitação postergada com **proposta masculina** (**feminina**) produz um escalonamento **ótimo-masculino** (**ótimo-feminino**).

# Emparelhamentos ótimos

Emparelhamento  $\nu$  é **ótimo-masculino** se não há emparelhamento estável  $\mu$  tq  $\mu(h) \succ_h \nu(h)$  ou  $\mu(h) = \nu(h)$  para todo  $h$  em  $H$  e  $\mu(j) \succ_j \nu(j)$  para algum  $j$  em  $H$ .

Definição de emparelhamento **ótimo-feminino** é análoga.

**Teorema:** O algoritmo da aceitação postergada com **proposta masculina** (**feminina**) produz um escalonamento **ótimo-masculino** (**ótimo-feminino**).

Provaremos algo mais forte!

## Dica para a prova anterior

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

## Dica para a prova anterior

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .  
(Pense que as propostas podem estar sendo feitas assincronamente.)

## Dica para a prova anterior

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .  
(Pense que as propostas podem estar sendo feitas assincronamente.)

Em particular,  $S$  é um emparelhamento!

## Dica para a prova anterior

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .  
(Pense que as propostas podem estar sendo feitas assincronamente.)

Em particular,  $S$  é um emparelhamento!

Se provarmos o acima, provamos o teorema anterior:

## Dica para a prova anterior

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .  
(Pense que as propostas podem estar sendo feitas assincronamente.)

Em particular,  $S$  é um emparelhamento!

Se provarmos o acima, provamos o teorema anterior:

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

## Afirmação mais forte

$S = \{(h, \text{best}(h)) : h \in H\}$ , onde  $\text{best}(h)$  é mulher melhor posicionada na lista de  $h$  que é esposa de  $h$  num emparelhamento estável.

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

## Afirmação mais forte

$S = \{(h, \text{best}(h)) : h \in H\}$ , onde  $\text{best}(h)$  é mulher melhor posicionada na lista de  $h$  que é esposa de  $h$  num emparelhamento estável.

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Esboço da prova:** Seja  $E$  execução do algoritmo que não devolve  $S$ . Então existe um  $h$  que terminou com mulher distinta de  $m = \text{best}(h)$ . Note que  $h$  fez uma proposta para  $m$  e foi rejeitado. Suponha que foi o primeiro  $h$  que foi rejeitado em  $E$  por  $\text{best}(h)$ . Seja  $h'$  o homem com quem  $m$  estava quando rejeitou  $h$ . Então  $h' \succ_m h$ .

Seja  $S'$  um emparelhamento estável em que  $h$  está casado com  $m$  e seja  $m'$  a esposa de  $h'$  em  $S'$ . Como, em  $E$ ,  $h'$  não tinha sido rejeitado por  $\text{best}(h')$  quando estava com  $m$ , temos que  $m \succeq_{h'} \text{best}(h')$ . Por outro lado,  $\text{best}(h') \succeq_{h'} m'$ , pois  $h'$  terminou com  $m'$  em  $S'$ .

Logo  $m \succ_{h'} m'$ , e portanto  $(h', m)$  é um par bloqueador para  $S'$ .  $\square$

## Resumo e exercício

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

## Resumo e exercício

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) produz um escalonamento ótimo-masculino (ótimo-feminino).

**Exercício:** Defina  $\text{worst}(m)$  analogamente e mostre que, no algoritmo da proposta masculina, cada mulher  $m$  fica casada com  $\text{worst}(m)$ .

# Algoritmo à prova de estratégia

Não há incentivo para alguém mentir sua ordem de preferência.

## Algoritmo à prova de estratégia

Não há incentivo para alguém mentir sua ordem de preferência.

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

# Algoritmo à prova de estratégia

Não há incentivo para alguém mentir sua ordem de preferência.

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) é um mecanismo à prova de estratégia para os homens (mulheres) .

# Algoritmo à prova de estratégia

Não há incentivo para alguém mentir sua ordem de preferência.

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) é um mecanismo à prova de estratégia para os homens (mulheres) .

Óbvio da afirmação acima.

# Algoritmo à prova de estratégia

Não há incentivo para alguém mentir sua ordem de preferência.

Seja  $\text{best}(h)$  a mulher mais bem posicionada na lista de  $h$  tq existe emparelhamento estável com  $h$  casado com  $\text{best}(h)$ .

Considere  $S = \{(h, \text{best}(h)) : h \in H\}$ .

Qualquer que seja a ordem em que os homens proponham no algoritmo da proposta masculina, o emparelhamento produzido é exatamente  $S$ .

**Teorema:** O algoritmo da aceitação postergada com proposta masculina (feminina) é um mecanismo à prova de estratégia para os homens (mulheres) .

Óbvio da afirmação acima.

Vimos na aula que é possível, em alguns casos, uma mulher terminar com um marido melhor mentindo a sua ordem de preferência.