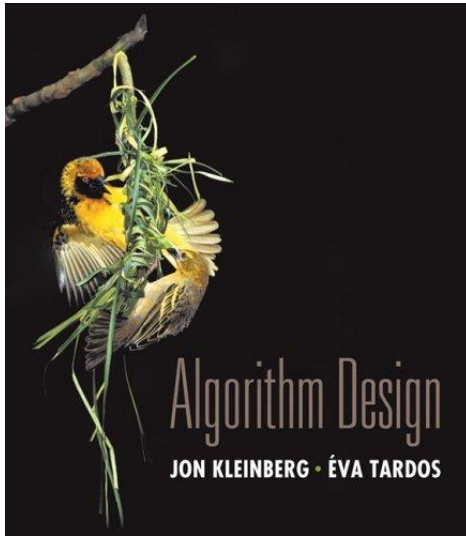




# SCC – 5900

## Projeto de Algoritmos

### O Casamento Estável



João Batista  
ICMC-USP

# Um Primeiro Problema: Casamento Estável

- Em 1962, David Gale e Lloyd Shapney, economistas matemáticos, se propuseram a resolver o seguinte problema:
- Será possível construir um sistema (admissão em uma Universidade, processo de recrutamento de empregados por empresas) de forma que este seja auto-regulável (self-enforcing)?
- Entrada: a lista de preferências entre candidatos e empregadores (universitários e universidades, por exemplo)
- Um candidato tem uma ordem de preferência por companhias.
- As companhias, as receber as inscrições, formam uma ordem de chamada para os candidatos.
- O que pode dar errado com este sistema???

# Um Primeiro Problema: Casamento Estável

- Suponha que alunos possam trocar de ofertas quando bem entenderem.
- Imagine que as empresas possam dispensar alguém previamente acordado, e contratar outros que dispensaram a oferta de estágio por ter escolhido outra empresa?
- Da maneira como formulada, a relação pode terminar de forma caótica.
- Existe estabilidade neste processo?
- Se não existe, não é possível dizer que o processo é auto-ajustável. Por que?
  - Porque as pessoas agem em interesse próprio e o sistema acaba entrando em colapso !

# Casamento de Candidatos com Empresas

**Objetivo.** Dado um conjunto de preferências entre candidatos e empresas, desenvolva um processo de admissão **auto-regulável**. (self-enforcing)

**Par instável:** candidato  $x$  e empresa  $y$  são **instáveis** se:

- $x$  prefere  $y$  à empresa a qual foi atribuído.
- $y$  prefere  $x$  a um dos seus candidatos admitidos.
- Do ponto de vista econômico, relações instáveis trazem muitos problemas, inviabilizando qq planejamento decente.

**Atribuição estável:** Atribuição sem pares instáveis.

- Condição natural e desejável.
- Interesses pessoais impedirão qualquer acordo candidato/empresa de ser feito.

# Um Primeiro Problema: Casamento Estável

- Qual seria uma solução ideal para este problema?
  - Exigir altruísmo tanto de empresa quanto de empregados e esperar que estes mantenham suas escolhas, mesmo que prefiram a outros?
- Não
  - O ideal é a situação em que o interesse próprio (tanto de empresa quanto de empregado) previna que ofertas de empregos sejam desfeitas e redirecionadas
  - Exemplo: se alguém que trabalha para X, liga para a firma Y, se oferecendo para abandonar X e ir para Y, Y deveria dizer: sinto muito... com base nas escolhas prévias que fizemos, nós não temos preferência em lhe contratar. Obrigado !
  - O mesmo valeria para empresas indo atrás de candidatos já empregados por outras empresas, mas não obtendo sucesso (os empregados já estão contentes onde estão!)

# Problema do Casamento Estável

**Objetivo.** Dado  $n$  homens (H) e  $n$  mulheres (M), encontre um casamento “adequado”.

- Cada homem faz uma lista de preferências de mulheres.
- Cada mulher faz uma lista de preferências de homens.

	favorito ↓ 1°	2°	menos favorito ↓ 3°
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

*Lista de preferência dos homens*

	favorito ↓ 1°	2°	menos favorito ↓ 3°
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

*Lista de preferência das mulheres*

Casamento S: conjunto de pares ordenados (H X M) em que cada membro de H e M aparece em no máximo um par em S. Atenção: no máximo significa que alguém pode ficar ser par !!!

# Problema do Casamento Estável

**Casamento perfeito ( $S'$ ):** não há poligamia e “solteirice”.

- Cada homem casa exatamente com uma mulher.
- Cada mulher casa exatamente com um homem.
- Cada membro  $H$  e  $M$  aparece em exatamente um par em  $S'$ .

**Estabilidade:** não há incentivo para que um par prejudique outros casamentos.

- No casamento  $S$ , um par não casado  $m-w$  é **instável** se o homem  $m$  e a mulher  $w$  preferir um ao outro ao invés de seus atuais pares.
- Pares instáveis  $m-w$  poderiam se separar de seus atuais pares.

**Casamento estável:** casamento perfeito sem pares instáveis.

**Problema do Casamento Estável.** Dado um lista de preferências de  $n$  homens e  $n$  mulheres, encontre um casamento estável, *se ele existir*.

# Problema do Casamento Estável

P. A atribuição X-C, Y-B, Z-A é estável?

	favorit o ↓		menos favorito
	1°	2°	3°
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

*Lista de preferência dos homens*

	favorit o ↓		menos favorito
	1°	2°	3°
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

*Lista de preferência das mulheres*



# Problema do Casamento Estável

P. A atribuição X-C, Y-B, Z-A é estável?

R. Não. Bertha e Xavier irão se unir.

	favorito ↓		menos favorito
	1°	2°	3°
Xavier	Amy	<b>Bertha</b>	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

*Lista de preferência dos homens*

	favorito ↓		menos favorito
	1°	2°	3°
Amy	Yancey	Xavier	Zeus
Bertha	<b>Xavier</b>	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

*Lista de preferência das mulheres*

# Problema do Casamento Estável

Q. A atribuição X-A, Y-B, Z-C é estável?

R. Sim.

	favorito ↓		menos favorito
	1°	2°	3°
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

*Lista de preferência dos homens*

	favorito ↓		menos favorito
	1°	2°	3°
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

*Lista de preferência das mulheres*

# Problema do Companheiro de Quarto Estável

Q. Sempre existirão casamentos estáveis?

R. Não é óbvio a priori.

Mas a resposta é sim. É possível mostrar que para cada conjunto de preferências entre mulheres e homens existirá um casamento estável. E também existe um algoritmos eficiente que faz isso.

Infos importantes:

a) todos estão solteiros no início

b) Em 1962 a sociedade era ainda mais machista e o algoritmo foi projetado de forma que os homens tomavam inicialmente a iniciativa de escolher as mulheres em sua lista de preferência. Chamaremos esta atitude de: pedir em casamento (*propose* (m,w))

c) Inicialmente, é arriscado para w rejeitar o pedido de m. Então dizemos que ao fazer a proposta, teremos um par intermediário (m,w) em noivado (*engagement*)

d) No entanto, futuramente, se m' pedir a mão de w (e w preferir m'), o par (m,w) será desfeito.

# Algoritmo de Proposta-e-Rejeição

Algoritmo de Proposta-e-Rejeição. [Gale-Shapley 1962] Método intuitivo que garante encontrar um casamento estável.

```
Initialize each person ( $m \in M$  ,  $h \in H$ ) to be free.
while (there is a man who is free and hasn't proposed to every woman) {
    Choose such a man m
    w = 1° woman on m's list to whom m has not yet proposed
    if (w is free)
        m and w become engaged
    else // w is currently engaged to m'
        if (w prefers m to her fiancé m')
            m and w become engaged;
            m' becomes free
        else w rejects m // m remains free!
}
```

Return the set S of engaged pairs (stable matching) !

# Prova de Corretude: Limite (complexidade)

**Observação 1.** Homens propões às mulheres em ordem decrescente de preferências.

**Observação 2.** Uma vez que uma mulher se compromete, ela não se torna solteira novamente, ela somente “troca para melhor”.

**Observação 3.** O homem, por sua vez pode voltar a ficar solteiro. E sua situação apenas “piora” com as trocas.

**Afirmção.** O algoritmo termina depois de no máximo  $n^2$  iterações do laço (while).

**Prova.** Cada iteração do laço um homem propõe casamento a uma mulher. Existem somente  $n^2$  possibilidades de pares de m e w !!!

	1°	2°	3°	4°	5°
Victor	A	B	C	D	E
Wyatt	B	C	D	A	E
Xavier	C	D	A	B	E
Yancey	D	A	B	C	E
Zeus	A	B	C	D	E

	1°	2°	3°	4°	5°
Amy	W	X	Y	Z	V
Bertha	X	Y	Z	V	W
Clare	Y	Z	V	W	X
Diane	Z	V	W	X	Y
Erika	V	W	X	Y	Z

# Prova de Corretude: casamento perfeito (não necessariamente estável)

**Afirmção.** Todos os homens e mulheres casam.

Se um homem está solteiro em algum momento da execução, é porque existe uma mulher a quem ele ainda não pediu em casamento.

**Prova.** (por contradição)

- Suponha que Zeus está solteiro, mas já propôs a todas as mulheres.
- Neste ponto cada uma das  $n$  mulheres está comprometida, por conta da observação 2 (uma vez comprometida, uma mulher não fica solteira novamente; apenas troca de marido) !!
- Como um conjunto de pares comprometidos representa um casamento, temos que ter  $n$  homens comprometidos neste ponto!
- Mas Zeus ainda está solteiro, o que é uma contradição !! ■

# Prova de Corretude: Estabilidade

**Afirmção.** Não há pares instáveis em S

**Prova.** (por contradição)

- Assuma que há instabilidade em S. Por exemplo  $(m, w)$  e  $(m', w')$  para a propriedade
  - $m$  prefere  $w'$  à  $w$  .....  $w'$  prefere  $m$  à  $m'$
- Na execução que produziu S, o último pedido de casamento de  $m$  foi para  $w$ . Perguntamos:
  - Será que  $m$  pediu a mão de  $w'$  em primeiro lugar na execução de S???
    - Se não, então ele prefere  $w$ , o que é uma contradição!
    - Se sim, então então ele foi rejeitado por  $w'$ , para ficar com um  $m''$ . Mas  $m'$  é o parceiro final de  $w'$ . Então ou  $m'' = m'$  ou (obs 2)  $w'$  prefere seu parceiro final  $m'$  à  $m''$ .  
Contradição !!
- Portanto, S é estável!! ▪

# Resumo

**Problema do Casamento Estável.** Dado  $n$  homens e  $n$  mulheres, e suas preferências, encontre um casamento estável se existir um.

**Algoritmo Gale-Shapley.** Garantias para encontrar um casamento estável para **qualquer** instância problema.

Q. Se existirem múltiplos casamentos estáveis, qual deles o algoritmo GS encontrará?

Q. Como implementar eficientemente o algoritmo GS?



# Entendendo a Solução

Q. Para uma dada instância do problema, pode haver vários casamentos estáveis. Todas as execuções do algoritmo GS produziram o mesmo casamento estável? Caso afirmativo, qual deles?

Seja a seguinte lista de preferências:

- m prefere w à w'
  - m' prefere w' a w
  - w prefere m' a m
  - w' prefere m a m'
- 
- Qual resultado teremos ao executarmos o alg. GS ?
  - Haveria outra solução possível e estável ??

# Entendendo a Solução

**Atribuição ótima para homens.** Cada homem casa com sua melhor parceira válida.

**Afirmção.** Todas as execuções de GS produzirão a **atribuição ótima para homens**, a qual é um casamento estável!

- . O algoritmo GS é altamente injusto para com as mulheres, pois são os homens que fazem o pedido!
- . Se fosse o contrário, seria injusto para com os homens.

# Resumo do Casamento Estável

**Problema do Casamento Estável.** Dado um conjunto de preferências de  $n$  homens e  $n$  mulheres, encontre um casamento e **estável**.

↖ nenhum homem ou mulher prefere estar com outro par fora o que foi atribuído.

**Algoritmo Gale-Shapley.** Encontre um casamento estável em tempo  $O(n^2)$ .

**Ótimo para Homens.** Na versão do algoritmo GS onde os homens propõe casamento, cada homem casa com sua melhor parceira válida.

↑  
 $w$  é um parceiro válido de  $m$  se existir algum casamento estável onde  $m$  e  $w$  são pares.

**Q.** O caso ótimo para os homens vem às custas da escolha das mulheres?

# Implementação Eficiente

**Implementação Eficiente.** Nós descrevemos uma implementação  $O(n^2)$ .

**Representando homens e mulheres.**

- Assuma que os homens são rotulados  $1, \dots, n$ .
- Assuma que as mulheres são rotuladas  $1', \dots, n'$ .

**Casamentos.**

- Mantenha uma lista de homens livres, e.g., em uma fila.
- Mantenha dois vetores `wife[m]`, e `husband[w]`.
  - atribua a entrada 0 se não há casamento
  - se  $m$  casar com  $w$  então `wife[m]=w` e `husband[w]=m`

**Homens propondo.**

- Para cada homem, mantenha uma lista de mulheres, ordenadas por preferência.
- Mantenha um vetor `count[m]` que conta o número de propostas feitas pelo homem  $m$ .

# Implementação Eficiente

## Mulheres rejeição/aceitação.

- Uma mulher  $w$  prefere um homem  $m$  ao homem  $m'$ ?
- Para cada mulher, crie uma lista **inversa** da lista de preferência dos homens.
- Tempo de acesso constante para cada consulta depois de um pré-processamento  $O(n)$ .

Amy	1°	2°	3°	4°	5°	6°	7°	8°
Pref	8	3	7	1	4	5	6	2

Amy	1	2	3	4	5	6	7	8
Inverse	4°	8°	2°	5°	6°	7°	3°	1°

```
for i = 1 to n  
  inverse[pref[i]] = i
```

Amy prefere o homem 3 ao 6  
desde que  $\text{inverse}[3] < \text{inverse}[6]$

2            7