

## Aula 1: Casamentos Estáveis

Instrutor: Berilhes Borges Garcia

Escriba: Felipe Thomaz Pedroni

## 1 Escopo do Curso

- Prova de Correção
- Análise do Consumo de Recursos
- Técnicas de Projeto
  - Exploração de Grafos
  - Algoritmos Gulosos
  - Dividir-e-Conquistar
  - Programação Dinâmica
  - Estruturação de Dados
  - Randomização
- NP- Completude e Intratabilidade
- Coisas que Você Deveria Conhecer
  - Lógica Proposicional
  - Combinatória Elementar (permutações, combinações, ...)
  - Teoria da Probabilidade Elementar
  - Estrutura de Dados Elementar (vetores, lista, pilhas, filas, etc)
  - Algoritmos de Ordenação Padrão (Insertion Sort, Quick Sort, Merge Sort, ...)
  - Tabelas Hash

## 2 Casamentos Estáveis

**Problema 1.** *Casamentos Estáveis: Um Exemplo Introdutório.*

Dado:

- $n$  mulheres  $m_1, m_2, \dots, m_n$ ;
- $n$  homens  $h_1, h_2, \dots, h_n$ ;
- $n$  casamentos  $(m_{i_1}, h_{j_1}), (m_{i_2}, h_{j_2}), \dots, (m_{i_n}, h_{j_n})$ .

*Exemplo:* Considere os casamentos entre 5(cinco) homens e 5(cinco) mulheres representados pela figura a seguir. Note que cada homem possui uma lista de preferência, o mesmo vale para as mulheres.

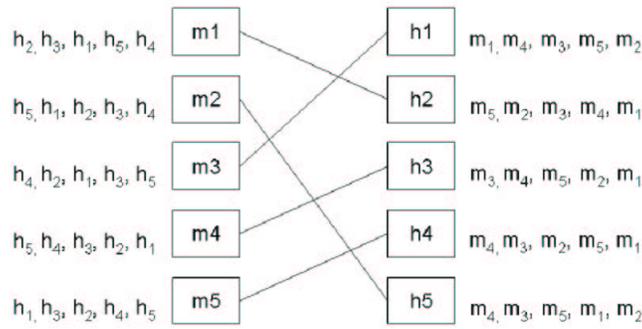


Figura 1: Casamentos e Listas de Preferências

**Definição 1.** Um casamento  $m_{i_k}, h_{j_k}$  é estável se:

- Não existe homem  $h_{j'}$ ,  $j' \neq j_k$ , cuja mulher  $m_{i_k}$  gosta mais do que  $m_{j_k}$
- $n$  homens  $h_1, h_2, \dots, h_n$ ;
- $n$  casamentos  $(m_{i_1}, h_{j_1}), (m_{i_2}, h_{j_2}), \dots, (m_{i_n}, h_{j_n})$ .

⊠

*Exemplo:* Casamentos Instáveis.

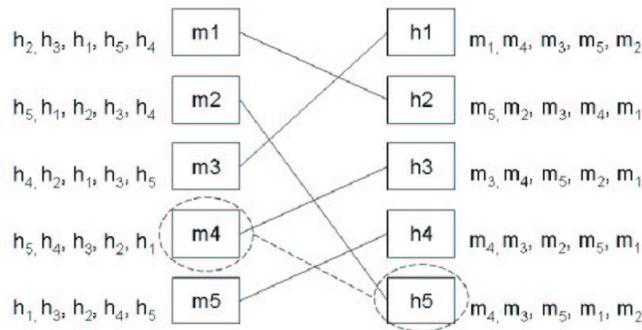


Figura 2: Casal Bloqueador  $(m_4, h_5)$

Note a existência de dois casamentos instáveis, pois  $m_4$  prefere  $h_5$  a seu atual marido ( $h_3$ ) e  $h_5$  prefere  $m_4$  a sua atual esposa ( $m_2$ ). Este casal  $(m_4, h_5)$  é denominado casal bloqueador ou casal desgarrado.

⊠

## 2.1 O Algoritmo de Gale-Shapley

**Algoritmo 1.** *Algoritmo Proposto* ( $H, M$ )

1. **while** existir um homem solteiro  $h$  (que não tenha feito propostas para todas as mulheres )  
**do**
2.      $h$  escolhe a sua mulher favorita  $m$  que ele não tenha cortejado antes
3.      $h$  propõe casamento à  $m$
4.     se  $m$  não é casada ou gosta mais de  $h$  do que seu atual parceiro  $h'$
5.         then  $m$  divorcia-se de  $h'$
6.          $m$  casa com  $h$

**Exercício 1.**

1. Existe sempre um conjunto  $n$  de casamentos estáveis?
2. O algoritmo sempre termina?
3. O algoritmo sempre produz a resposta correta?
4. Quão eficiente é o algoritmo? Nós podemos dar um limite superior para o seu tempo de execução?

## 2.2 Término do Algoritmo

**Lema 1.** *O Algoritmo Proposto termina depois de no máximo  $n^2$  iterações.*

- Existem  $n$  homens;
- Todo homem pode fazer propostas para  $n$  mulheres;
- Nenhum homem propõe para a mesma mulher duas vezes

Portanto, no máximo  $n^2$  propostas podem ser feitas. Em toda interação 1(uma) proposta é feita. Logo, o número total de iterações é, no máximo,  $n^2$ .

**Observação 1.** *Parar terminar, um algoritmo necessita fazer progresso regular em direção a uma solução.*

## 2.3 Todo Mundo Casa-se no Final

**Lema 2.** *Quando o Algoritmo Proposto termina, toda mulher contraiu casamento (e, portanto todo homem também). Prova por Contradição<sup>1</sup>*

**Suposição:** Existe uma mulher solteira  $m$  no fim do algoritmo.

- Uma mulher, um vez casada, fica casada (não necessariamente com o mesmo homem);
- Se existe uma mulher solteira  $m$ , deve existir um homem solteiro  $h$ .
- Quando o algoritmo termina,  $h$  deve ter feito proposta para todas as mulheres, incluindo  $m$ ;
- $m$  teria casado com  $h$ , então.

**CONTRADIÇÃO!!!**

## 2.4 Correção do Algoritmo

**Lema 3.** *Considere uma execução do Algoritmo Proposto que retorna um conjunto  $S$ . O conjunto  $S$  é um casamento estável*

**Prova:** Assuma que exista uma instabilidade com respeito a  $S$ , nosso objetivo será mostrar que isto é uma contradição. Como definido anteriormente, tal instabilidade envolverá dois pares,  $(h, m)$  e  $(h', m')$ , em  $S$  com as propriedades que:

- $h$  prefere  $m'$  à  $m$ , e
- $m'$  prefere  $h$  à  $h'$

Na execução do algoritmo que produziu  $S$ , a última proposta de  $h$  foi, por definição, para  $m$ . Nós, agora, perguntamos:  $h$  propôs à  $m'$  em algum ponto anterior desta execução? Se ele não propôs, então  $m$  ocorre em posição maior na lista de preferências de  $h$  que  $m'$ , contradizendo nossa suposição que  $h$  prefere  $m'$  à  $m$ . Se ele fez uma proposta, então ele foi rejeitado por  $m'$ , que prefere algum outro homem  $h''$ . Como  $h'$  é o parceiro final de  $m'$ , assim ou  $h'' = h'$ , ou  $m'$  prefere  $h'$  à  $h''$ , de qualquer forma isto contradiz nossa suposição que  $m'$  prefere  $h$  à  $h'$ .

---

<sup>1</sup>**Prova Por Contradição:**

- Assuma que o que nós queremos provar não é correto;
- Provar isto conduz a uma contradição ou com a suposição que nós fizemos ou com algum fato conhecido.