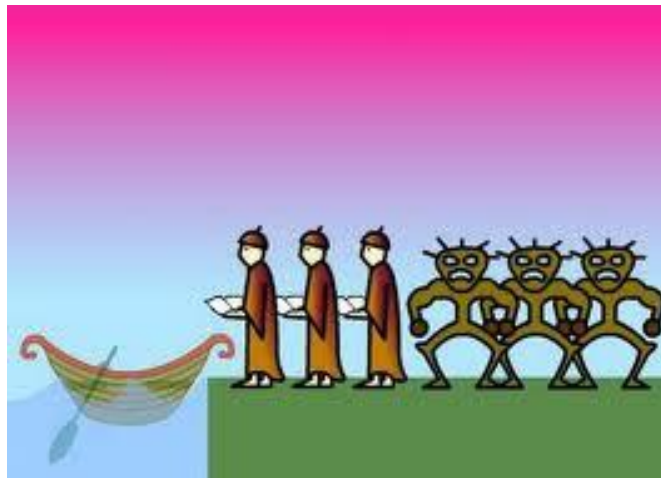


# Exercícios

- 3 missionários e 3 canibais
  - Construir um plano para levar todos para o outro lado do rio
  - 1 barco pode carregar 2
  - Missionários nunca devem estar em menor número que canibais





# Missionários e Canibais

---

- Estado  $\langle M, C, \text{PosiçãoDoBarco} \rangle$ 
  - M: número de missionários na margem esquerda do rio
  - C: número de canibais na margem esquerda do rio
  - PosiçãoDoBarco: posição do barco (MargemEsq, MargemDir)



# Missionários e Canibais

---

- Estado inicial:  $\langle 3, 3, \text{MargemEsquerda} \rangle$
- Objetivo:  $\langle 3, 3, \text{MargemDireita} \rangle$
- Operadores:  $\langle M, C \rangle$ 
  - M: número de missionários no barco
  - C: número de canibais no barco
  - Operadores válidos:  $\langle 1, 0 \rangle \langle 2, 0 \rangle \langle 1, 1 \rangle \langle 0, 1 \rangle \langle 0, 2 \rangle$



# Missionários e Canibais

- Todas as combinações possíveis

<3, 3, MargemEsq>

<2, 3, MargemEsq>

<1, 3, MargemEsq>

<3, 2, MargemEsq>

<2, 2, MargemEsq>

<1, 2, MargemEsq>

<3, 1, MargemEsq>

<2, 1, MargemEsq>

<1, 1, MargemEsq>

<0, 1, MargemEsq>

<0, 2, MargemEsq>

<0, 3, MargemEsq>

<3, 0, MargemEsq>

<2, 0, MargemEsq>

<1, 0, MargemEsq>

<0, 0, MargemEsq>

<3, 3, MargemDir>

<2, 3, MargemDir>

<1, 3, MargemDir>

<3, 2, MargemDir>

<2, 2, MargemDir>

<1, 2, MargemDir>

<3, 1, MargemDir>

<2, 1, MargemDir>

<1, 1, MargemDir>

<0, 1, MargemDir>

<0, 2, MargemDir>

<0, 3, MargemDir>

<3, 0, MargemDir>

<2, 0, MargemDir>

<1, 0, MargemDir>

<0, 0, MargemDir>

# Missionários e Canibais

- Contudo, algumas combinações não são válidas
  - Missionários nunca devem estar em menor número que canibais

<3, 3, MargemEsq>

<2, 3, MargemEsq>

<1, 3, MargemEsq>

<3, 2, MargemEsq>

<2, 2, MargemEsq>

<1, 2, MargemEsq>

<3, 1, MargemEsq>

<2, 1, MargemEsq>

<1, 1, MargemEsq>

<0, 1, MargemEsq>

<0, 2, MargemEsq>

<0, 3, MargemEsq>

<3, 0, MargemEsq>

<2, 0, MargemEsq>

<1, 0, MargemEsq>

<0, 0, MargemEsq>

<3, 3, MargemDir>

<2, 3, MargemDir>

<1, 3, MargemDir>

<3, 2, MargemDir>

<2, 2, MargemDir>

<1, 2, MargemDir>

<3, 1, MargemDir>

<2, 1, MargemDir>

<1, 1, MargemDir>

<0, 1, MargemDir>

<0, 2, MargemDir>

<0, 3, MargemDir>

<3, 0, MargemDir>

<2, 0, MargemDir>

<1, 0, MargemDir>

<0, 0, MargemDir>



# Missionários e Canibais

- Todas as combinações possíveis e válidas

<3, 3, MargemEsq>

<3, 3, MargemDir>

<3, 2, MargemEsq>

<3, 2, MargemDir>

<2, 2, MargemEsq>

<2, 2, MargemDir>

<3, 1, MargemEsq>

<3, 1, MargemDir>

<1, 1, MargemEsq>

<1, 1, MargemDir>

<0, 1, MargemEsq>

<0, 1, MargemDir>

<0, 2, MargemEsq>

<0, 2, MargemDir>

<0, 3, MargemEsq>

<0, 3, MargemDir>

<3, 0, MargemEsq>

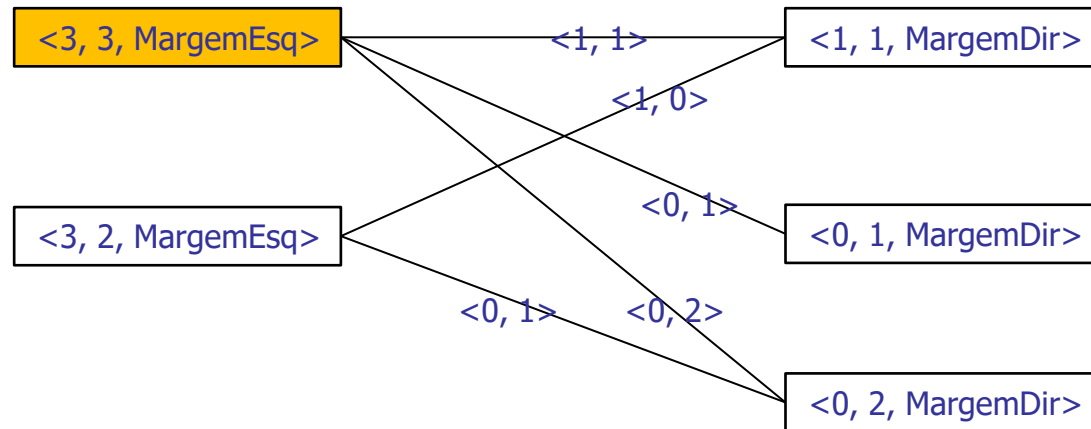
<3, 0, MargemDir>

<0, 0, MargemEsq>

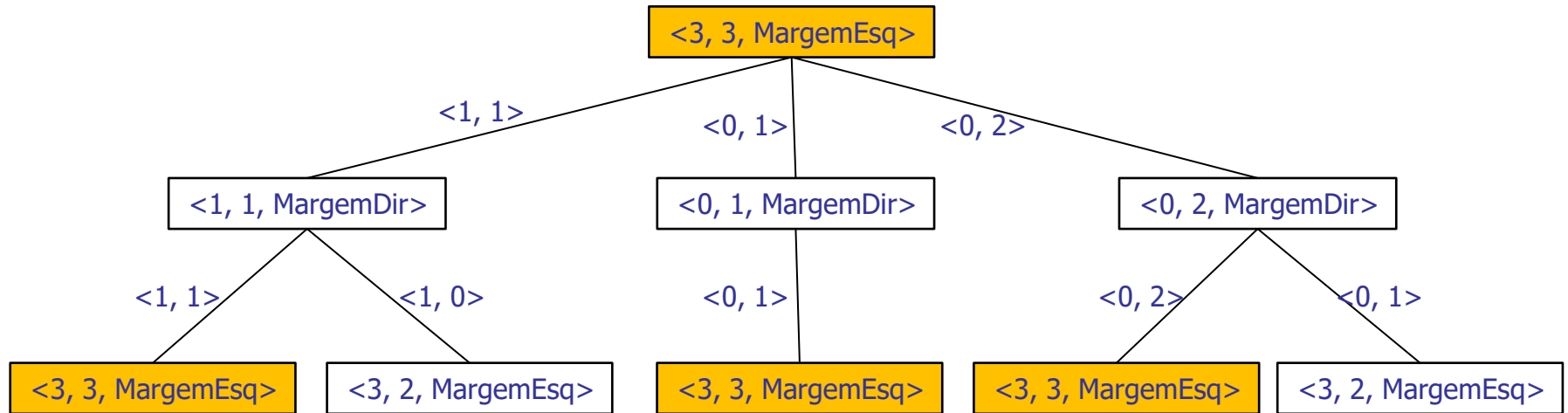
<0, 0, MargemDir>

# Missionários e Canibais

- Operadores válidos  $\langle M, C \rangle$ :  $\langle 1, 0 \rangle$   $\langle 2, 0 \rangle$   $\langle 1, 1 \rangle$   $\langle 0, 1 \rangle$   $\langle 0, 2 \rangle$



# Missionários e Canibais







# Processo de busca em largura

- Estado atual: (3, 3, MargemEsq)
- Novos estados gerados: [((1, 1, MargemDir), (1, 1)), ((0, 1, MargemDir), (0, 1)), ((0, 2, MargemDir), (0, 2))]
- Estado atual: (1, 1, MargemDir)
- Novos estados gerados: [((3, 2, MargemEsq), (1, 0)), ((3, 3, MargemEsq), (1, 1))]
- Estado atual: (0, 1, MargemDir)
- Novos estados gerados: [((3, 3, MargemEsq), (0, 1))]
- Estado atual: (0, 2, MargemDir)
- Novos estados gerados: [((3, 2, MargemEsq), (0, 1)), ((3, 3, MargemEsq), (0, 2))]
- Estado atual: (3, 2, MargemEsq)
- Novos estados gerados: [((1, 1, MargemDir), (1, 0)), ((0, 2, MargemDir), (0, 1)), ((0, 3, MargemDir), (0, 2))]
- Estado atual: (0, 3, MargemDir)
- Novos estados gerados: [((3, 1, MargemEsq), (0, 1)), ((3, 2, MargemEsq), (0, 2))]
- Estado atual: (3, 1, MargemEsq)
- Novos estados gerados: [((2, 2, MargemDir), (2, 0)), ((0, 3, MargemDir), (0, 1))]
- ...



# Processo de busca

---

- ...
- Estado atual: (2, 2, MargemDir)
- Novos estados gerados: [((3, 1, MargemEsq), (2, 0)), ((2, 2, MargemEsq), (1, 1))]
- Estado atual: (2, 2, MargemEsq)
- Novos estados gerados: [((3, 1, MargemDir), (2, 0)), ((2, 2, MargemDir), (1, 1))]
- Estado atual: (3, 1, MargemDir)
- Novos estados gerados: [((2, 2, MargemEsq), (2, 0)), ((0, 3, MargemEsq), (0, 1))]
- Estado atual: (0, 3, MargemEsq)
- Novos estados gerados: [((3, 1, MargemDir), (0, 1)), ((3, 2, MargemDir), (0, 2))]
- Estado atual: (3, 2, MargemDir)
- Novos estados gerados: [((1, 1, MargemEsq), (1, 0)), ((0, 2, MargemEsq), (0, 1)), ((0, 3, MargemEsq), (0, 2))]
- Estado atual: (1, 1, MargemEsq)
- Novos estados gerados: [((3, 2, MargemDir), (1, 0)), ((3, 3, MargemDir), (1, 1))]
- Estado atual: (0, 2, MargemEsq)
- Novos estados gerados: [((3, 2, MargemDir), (0, 1)), ((3, 3, MargemDir), (0, 2))]
- Estado atual: (3, 3, MargemDir)



# Processo de busca

---

- Plano para levar todos para o outro lado do rio
  - (1, 1)
  - (1, 0)
  - (0, 2)
  - (0, 1)
  - (2, 0)
  - (1, 1)
  - (2, 0)
  - (0, 1)
  - (0, 2)
  - (1, 0)
  - (1, 1)



# Trabalho individual I

---

Implementação de um algoritmo de busca para resolver o problema dos missionários e canibais



# Trabalho individual I

---

- Implementar o algoritmo de busca para resolver o problema dos Missionários e Canibais
  - Entregar o código do programa
    - Legível (identado, variáveis compreensíveis, etc), comentado (padrão JavaDoc), parametrizado (N missionários, N canibais, N operadores válidos) e orientado a objetos
  - Entregar o resultado do programa (print com o resultado)
  - Data de entrega: 07/06 (quarta)
    - Cópias ou programas similares serão avaliados com conceito zero