



# Inteligência Artificial

INTRODUÇÃO À ALGORITMOS DE BUSCA HEURÍSTICA

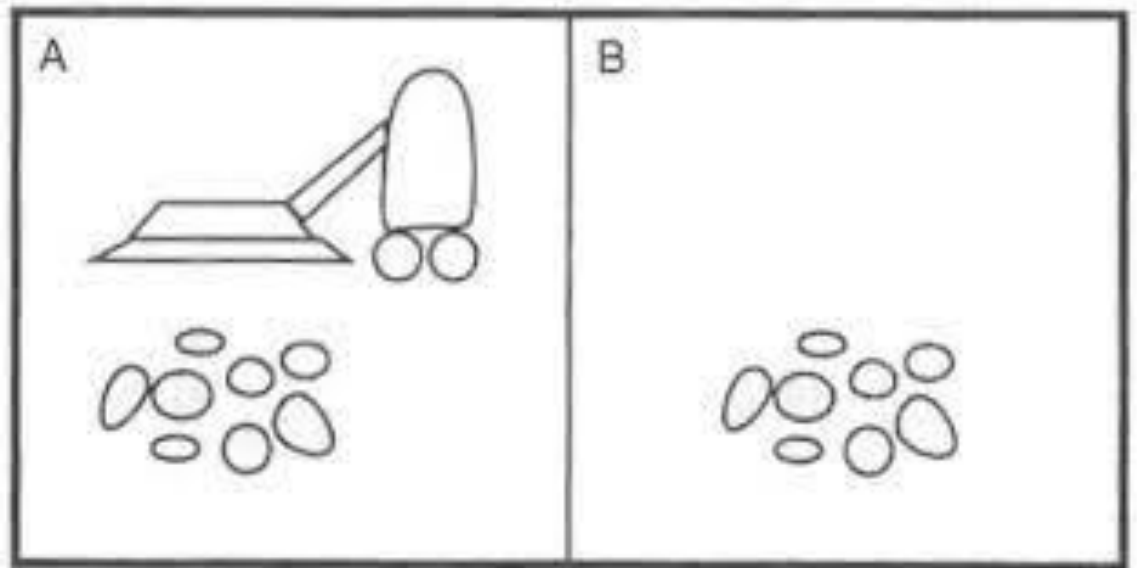
PROF. SÍLVIA MORAES

# Agentes e Algoritmos de Busca

- ▶ A tarefa de um agente baseado em objetivo é descobrir a sequência de ações que o levará a solução do problema (seu objetivo).
- ▶ O primeiro passo no caminho que leva à solução é a definição de uma abstração de mundo que capture apenas os elementos essenciais do problema.
- ▶ Vamos considerar inicialmente alguns miniproblemas cujos ambientes são completamente observáveis, discretos e determinísticos.

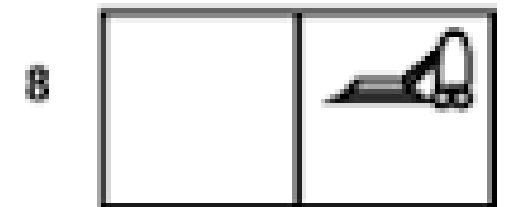
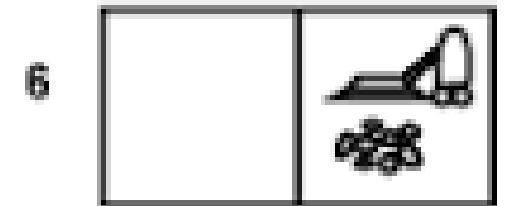
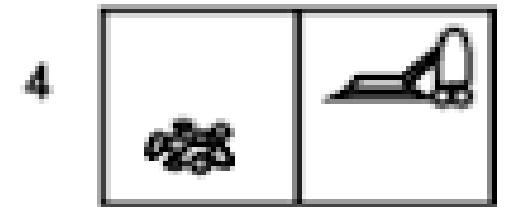
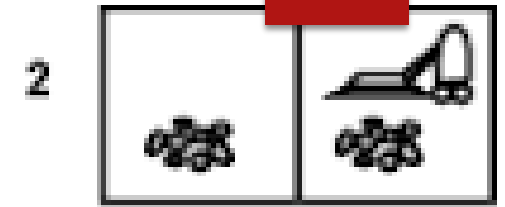
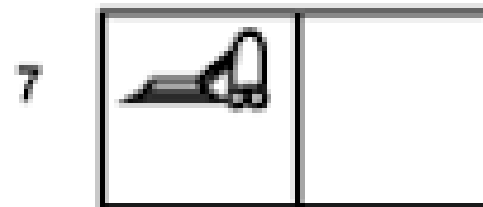
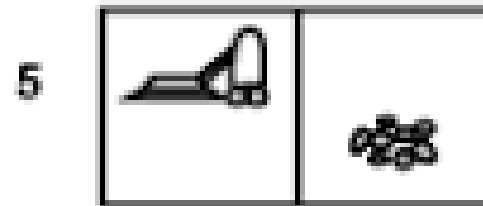
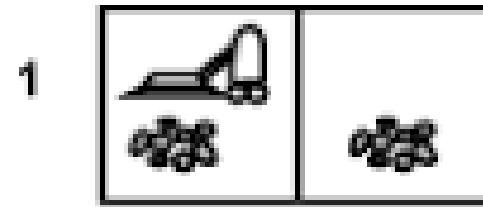
# Agentes e Algoritmos de Busca

- ▶ No caso do agente-exemplo Aspirador de pó, apresentado em aulas anteriores, que é capaz de:
  - ▶ Perceber em que local está;
  - ▶ Perceber se há sujeira nesse local;
  - ▶ Mover-se para esquerda
  - ▶ Mover-se para direita;
  - ▶ Aspirar sujeira ou
  - ▶ Não fazer nada.
- ▶ é suficiente criar uma representação abstrata:
  - ▶ [Posição, Estado de A, Estado de B]



# Estados de mundo

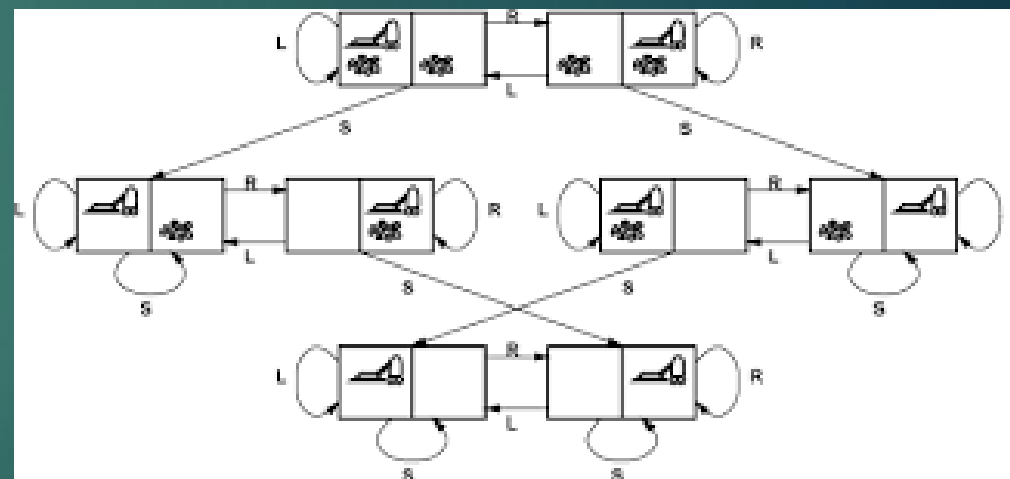
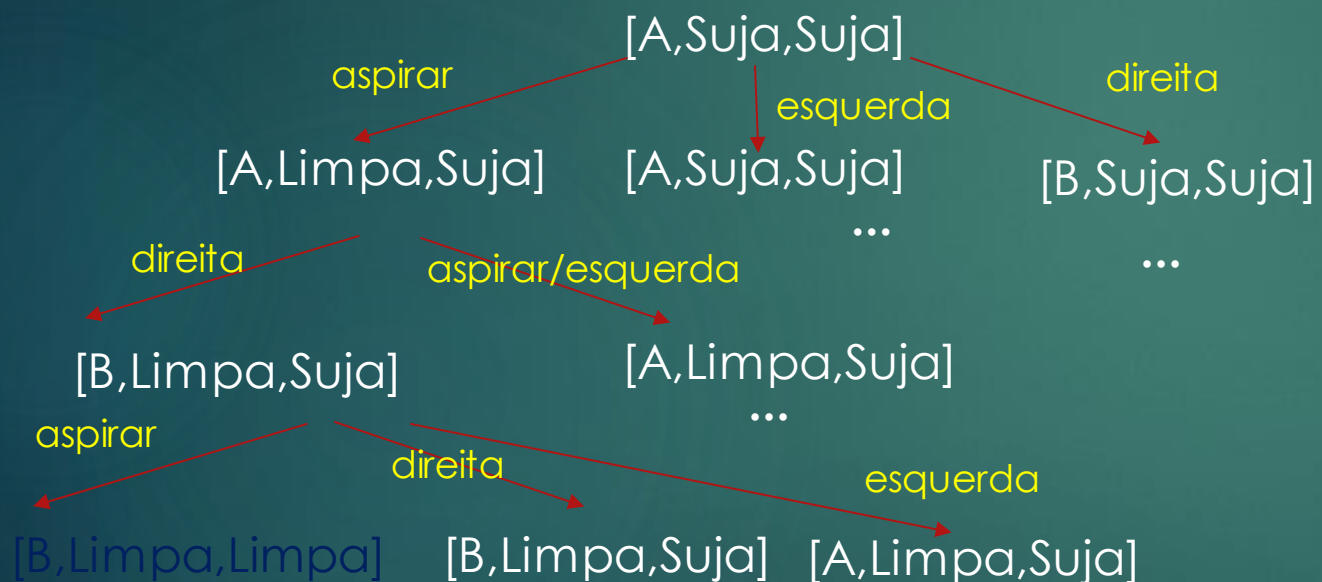
- ▶ Usando representação abstrata:
  - ▶ [Posição, Estado de A, Estado de B]
- ▶ Conseguimos representar os estados de mundo:
  - ▶ [A,Suja,Suja]
  - ▶ [A,Suja,Limpa]
  - ▶ [A,Limpa,Suja]
  - ▶ [A,Limpa,Limpa]
  - ▶ [B,Suja,Suja]
  - ▶ [B,Suja,Limpa]
  - ▶ [B,Limpa,Suja]
  - ▶ [B,Limpa,Limpa]



# Estado de Mundo

- ▶ No ambiente do Aspirador de pó é apenas por meio das ações do agente que o estado do mundo pode se modificar.
- ▶ Exemplo: (estadoAtual,ação,novoEstado)
  - ▶ ([A,Suja,Suja], **aspirar**, [A,Limpa,Suja])
  - ▶ ([A,Limpa,Suja], **direita**, [B,Limpa,Suja])
  - ▶ ([B,Limpa,Suja], **aspirar**, [B, Limpa,Limpa])
- ▶ Para o estado inicial [A,Suja,Suja], a sequência de ações procurada é **aspirar, direita, aspirar**

- ▶ O espaço de busca pode ser organizado na forma de uma árvore ou de um grafo



# Formalizando...

- ▶ Um problema pode ser definido formalmente por quatro componentes:
  - ▶ **estado inicial**: estado de mundo em que o agente começa a sua execução.
  - ▶ **descrição das ações**: define as ações possíveis do agente.
    - ▶ Em geral, usa uma função sucessor para gerar os estados válidos.
    - ▶ Dado um estado  $x$ , a função  $\text{sucessor}(x)$  retorna o conjunto de pares
  - ▶ **teste de objetivo**: usada para verificar se um dado estado é o estado objetivo.
  - ▶ **custo do caminho**: define o custo numérico a cada caminho que leva o estado inicial ao estado objetivo.

# Formalizando...

- ▶ Exemplo Aspirador de pó
  - ▶ **estado inicial**: pode ser qualquer um, tal como [A,sujo,sujo].
  - ▶ descrição das ações: **aspirar, direita e esquerda**.
    - ▶ **sucessor**([A,sujo,sujo]) = { (aspirar, [A,limpo,sujo]), (esquerda, [A,sujo,sujo]), (direita,[B,sujo,sujo]) }
  - ▶ teste de objetivo: verifica se todos os locais estão limpos, ou seja, se atingiu um dos estados **[A,limpo,limpo]** ou **[B,limpo,limpo]**.
  - ▶ custo do caminho: cada passo custa 1, logo é o número de passos do caminho.



# Algoritmo de Busca define o plano do agente

- ▶ A tarefa do agente é dentro do espaço de busca encontrar a sequência de ações que o leva do estado inicial ao estado objetivo.
- ▶ Para isso, são usados algoritmos de busca.
- ▶ No exemplo, o agente poderia encontrar os seguintes planos:
  - ▶ [aspirar, direita, aspirar] tem custo 3.
  - ▶ [direita, aspirar, esquerda, aspirar] tem custo 4

# Atividade I

- ▶ Para os estados inicial [B,Suja,Limpa] a seguir, quais os planos que o agente poderia encontrar ?

# Atividade II

- ▶ Se o problema fosse estendido para mais salas, da seguinte forma: 2 salas no térreo, 2 salas no primeiro piso, acesso às salas pelo corredor e um elevador para viabilizar o deslocamento entre os andares
- ▶ Sabendo que as operações disponíveis são:
  - ▶ Subir: do térreo para o primeiro piso desde que esteja no corredor
  - ▶ Descer: do primeiro piso para o térreo desde que esteja no corredor
  - ▶ Entrar na Sala 1: o agente deve estar no corredor do andar da Sala 1
  - ▶ Entrar na Sala 2: o agente deve estar no corredor do andar da Sala 2
  - ▶ Sair: o agente sai para o corredor do andar atual.
  - ▶ Aspirar: agente limpa a sala em que está posicionado.
- 1. Como representar ?
- 2. Como seria um plano para limpar todas as salas ?

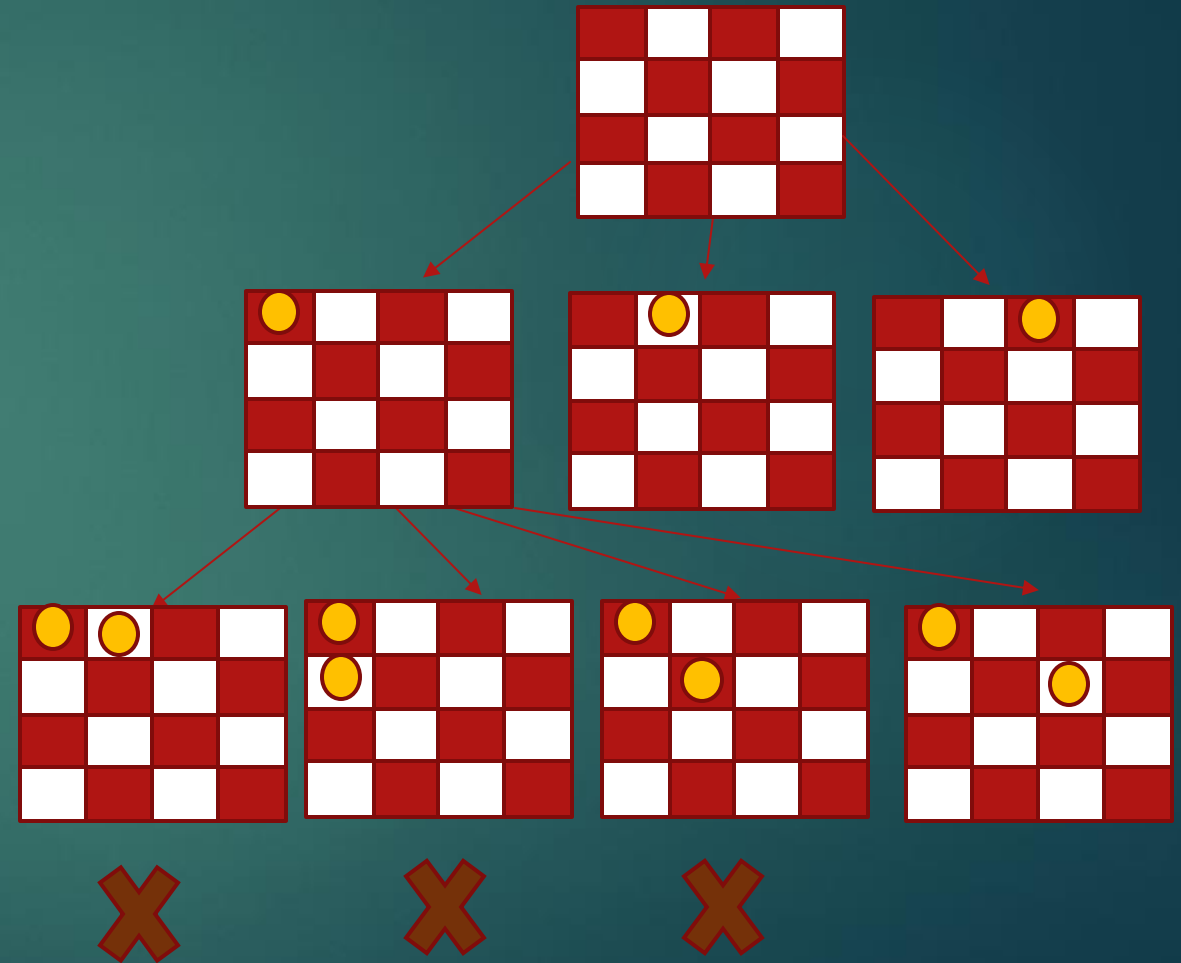
# Atividade III

- ▶ Considerando o estado atual do jogo da velha abaixo, monte o espaço de busca

▶	X	O	
▶		X	
▶			O

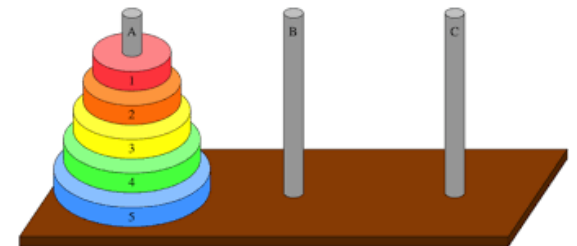
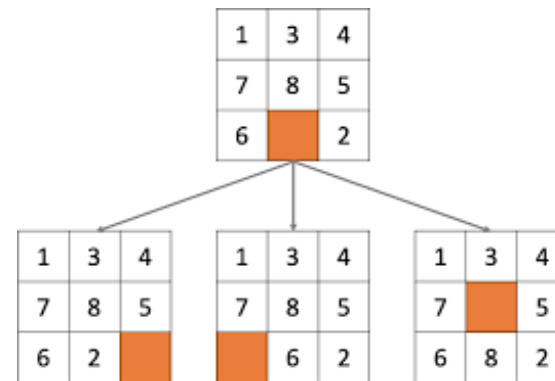
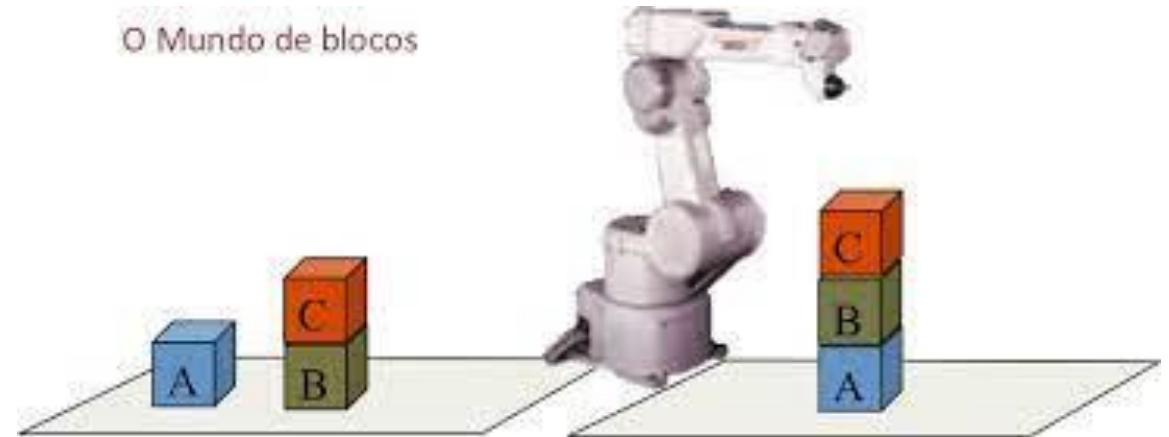
# Espaço de Busca

- ▶ Durante o processo de busca existe a possibilidade de desperçarmos tempo expandindo espaços que já foram encontrados e expandidos antes.
- ▶ Para alguns problemas, essa possibilidade nunca surge. O espaço de estados é uma árvore e só existe um caminho até cada estado. Ex: Problema das  $n$  rainhas.



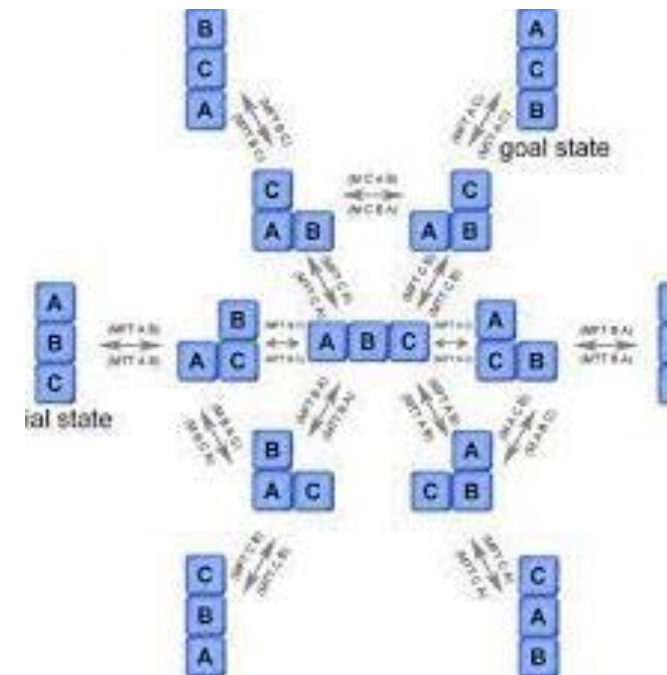
# Espaço de Busca

- ▶ Para outros, os estados repetidos são inevitáveis.
- ▶ Inclui problemas de localização de rotas ou quebra-cabeças deslizantes.
- ▶ As árvores de busca para esses problemas são infinitas (presença de ciclos).



# Espaço de Busca

- ▶ O que fazer ?
  - ▶ Podar alguns estados para reduzir a árvore a um tamanho finito (ex. profundidade máxima fixa).
  - ▶ Memorizar os estados visitados. Relação inversamente proporcional espaço x tempo. “Algoritmos que esquecem sua história estão condenados a repeti-la”.
  - ▶ Em problemas com muitos estados repetidos, **Busca-Em-Grafo** é mais eficiente do que **Busca-Em-Árvores**



# Algoritmos de Busca Informada

- ▶ A busca em um espaço de estados é o processo de procurar um caminho de solução iniciando no estado inicial até alcançar o estado objetivo.
- ▶ Algoritmos de Busca
  - ▶ sem informação (ou busca cega)
    - ▶ Busca em largura ou amplitude ou extensão
    - ▶ Busca em profundidade
  - ▶ com informação ou informada (usam funções heurísticas)



# Algoritmos de Busca Informada

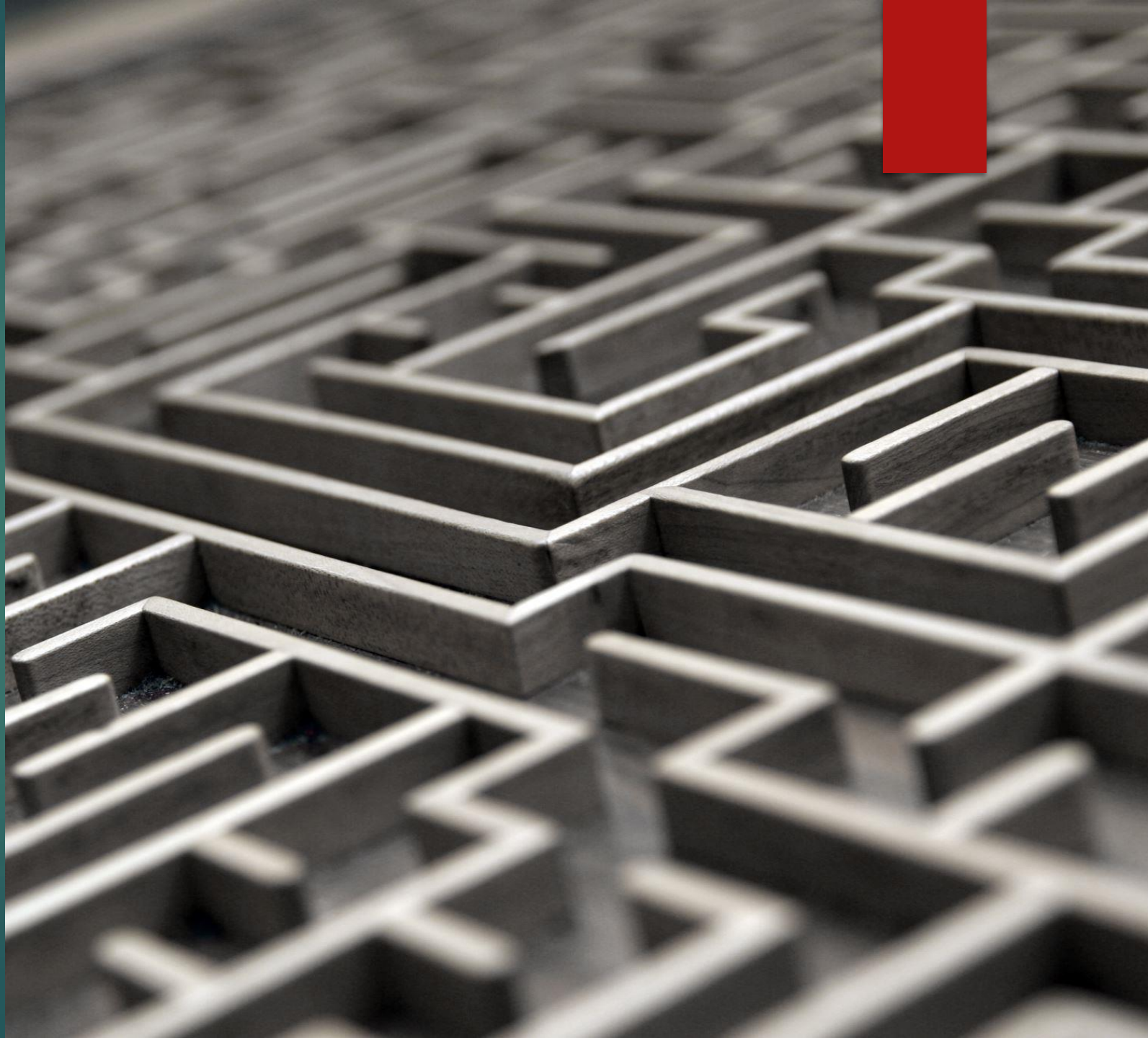
- ▶ Busca com informação (ou heurística) utiliza conhecimento específico do problema para encontrar a solução. Pode ser mais eficiente que a busca cega, pois não visita todos os nodos.
- ▶ Usa uma função de avaliação, heurística:  $f(n) = h(n)$
- ▶ Heurística:
  - ▶ probabilidade ou suposição a respeito da resolução de um problema
  - ▶ regra para escolher aqueles ramos em um espaço de estado que têm maior probabilidade de levarem a uma solução aceitável.

# Algoritmos de Busca Informada

- ▶ Razões para usá-la:
  - ▶ O problema não tem uma solução exata por causa das ambiguidades na formalização do problema ou nos dados disponíveis. Ex: diagnóstico médico
  - ▶ O problema tem solução exata, mas o custo computacional é proibitivo. Ex: jogo de xadrez

# Algoritmos de Busca Informada

- ▶ Algoritmo A\*, IDA\*
- ▶ Hill Climbing
- ▶ Simulated Annealing
- ▶ Algoritmos Genéticos



# Atividade IV

- ▶ Para responder as questões abaixo, consulte o capítulo 3 do livro do Russel e Norvig.
- 1. Qual a diferença entre um algoritmo de busca em extensão de um algoritmo de custo uniforme ?
- 2. Qual a diferença entre um algoritmo de busca em profundidade de um algoritmo com busca limitada ou iterativa ? Todos podem alcançar a solução ótima ?
- 3. Como funciona uma busca bidirecional ? Ela é sempre aplicável ? O que é necessário para aplicá-la ?
- 4. Quem consome mais memória os algoritmos de busca em extensão ou em profundidade ?

# Atividade V

- ▶ Considerando que o estado final desejado seja os números ordenados, a partir da primeira célula, pense em funções heurísticas que poderiam ajudar a escolher o estado seguinte.

