

FIAP

FIAP GRADUAÇÃO

**Engenharia de
Computação**

Inovação e Inteligência artificial e computacional

Prof^a. Thais Rodrigues Neubauer
profthais.neubauer@fiap.com.br

2020

BUSCAS

Referências base deste material:

- Russel, S.; Norvig, P. Inteligência Artificial. 2^a. edição. Editora Campus, 2004. Capítulos 3 e 4.
- Koza, J. R. Genetic Programming. MIT Press, 1992. Capítulo: 3.
- Linden, R. Algoritmos Genéticos. Brasport, 2006. Capítulos: 4, 5, 6, 7, e 9.



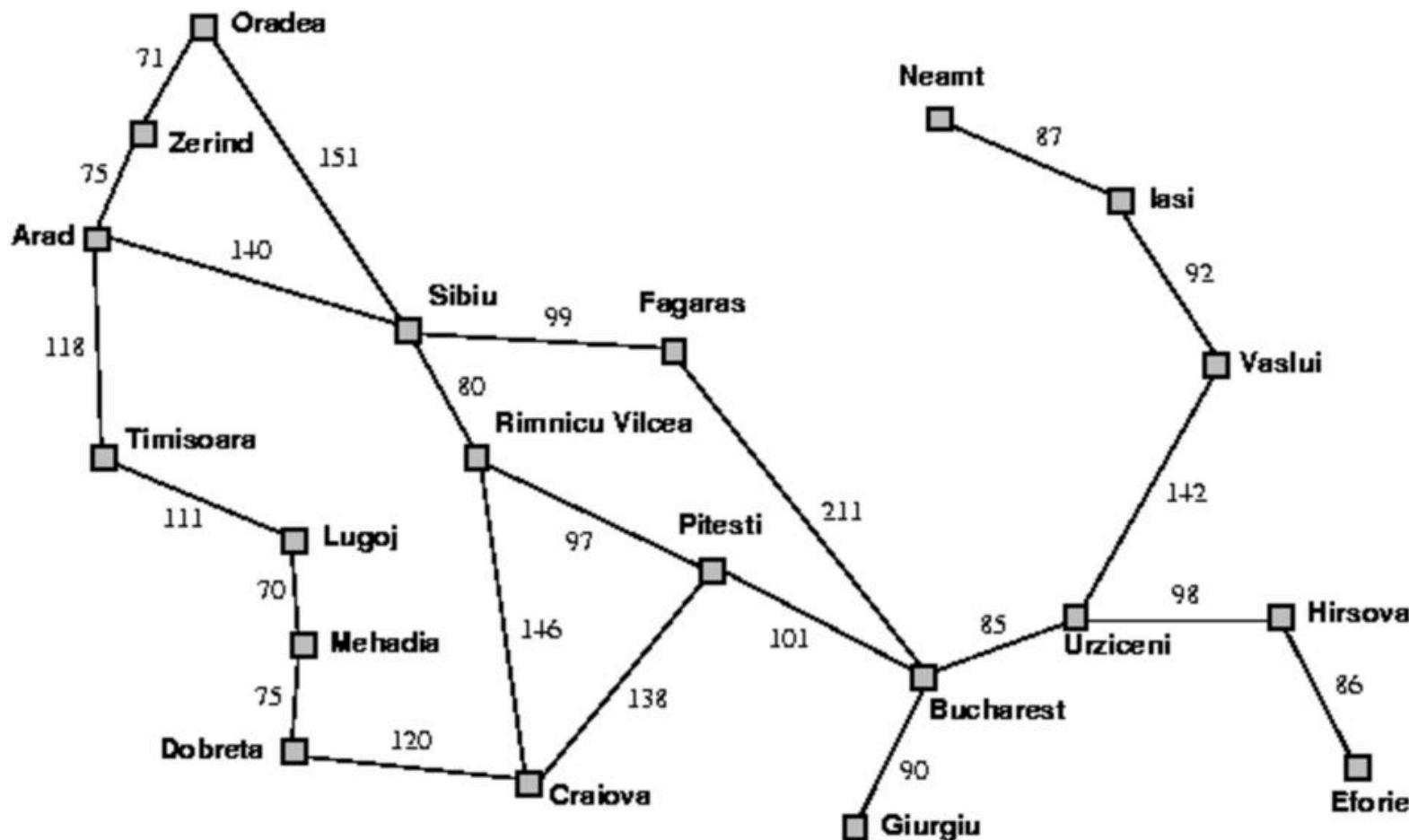
Representação – definição do problema:

- *Estado inicial*
- *Ações possíveis*
- *Teste de objetivo*
- *Função de custo* – custo numérico da solução

ESPAÇO DE BUSCA = ESTADO INICIAL + REALIZAÇÃO DAS AÇÕES POSSÍVEIS

BUSCAS – Exemplo: Chegar em Bucareste

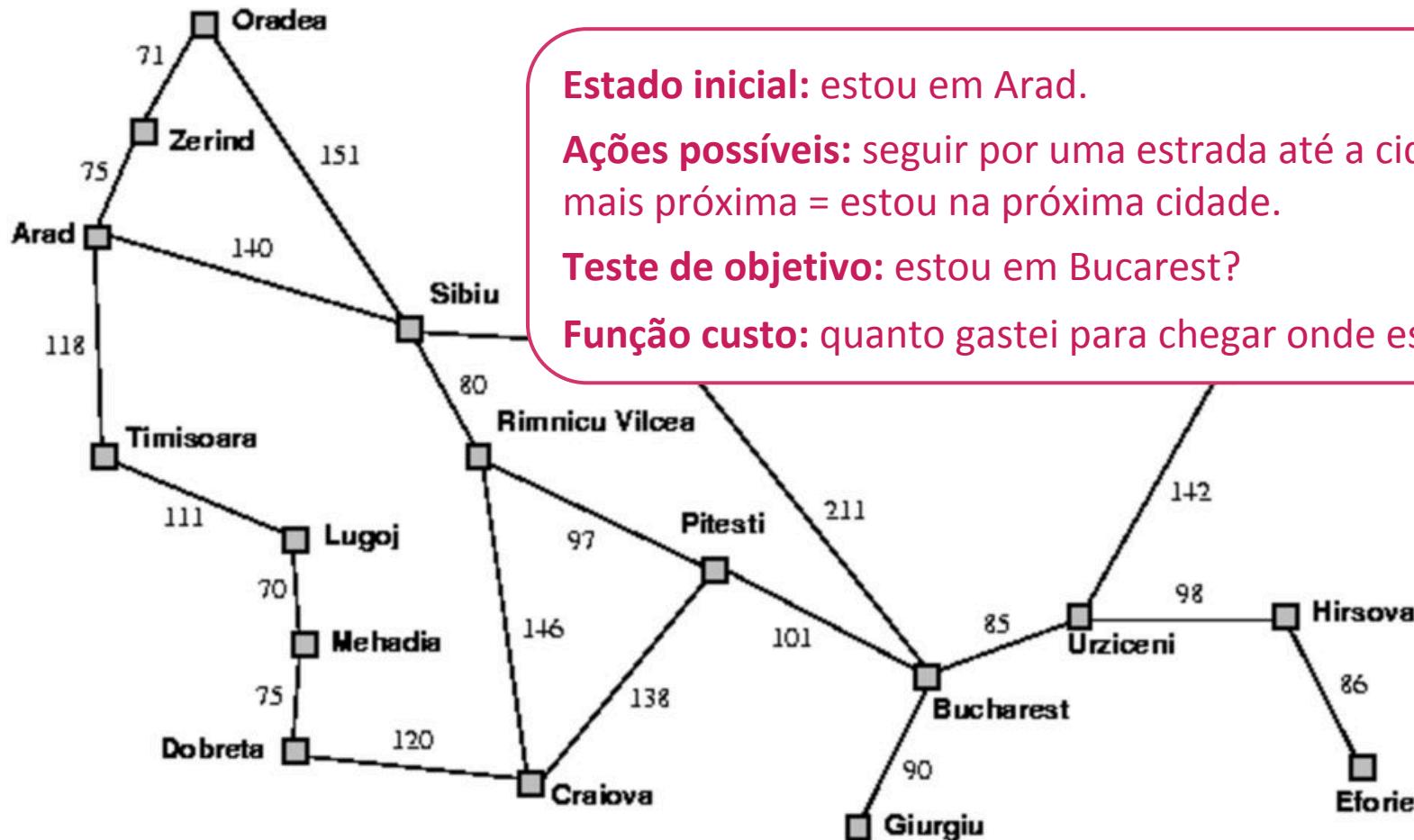
FIAP



Problema: CHEGAR EM BUCARESTE
Resultado da busca: o CAMINHO

BUSCAS – Exemplo: Chegar em Bucareste

FIAP



Estado inicial: estou em Arad.

Ações possíveis: seguir por uma estrada até a cidade mais próxima = estou na próxima cidade.

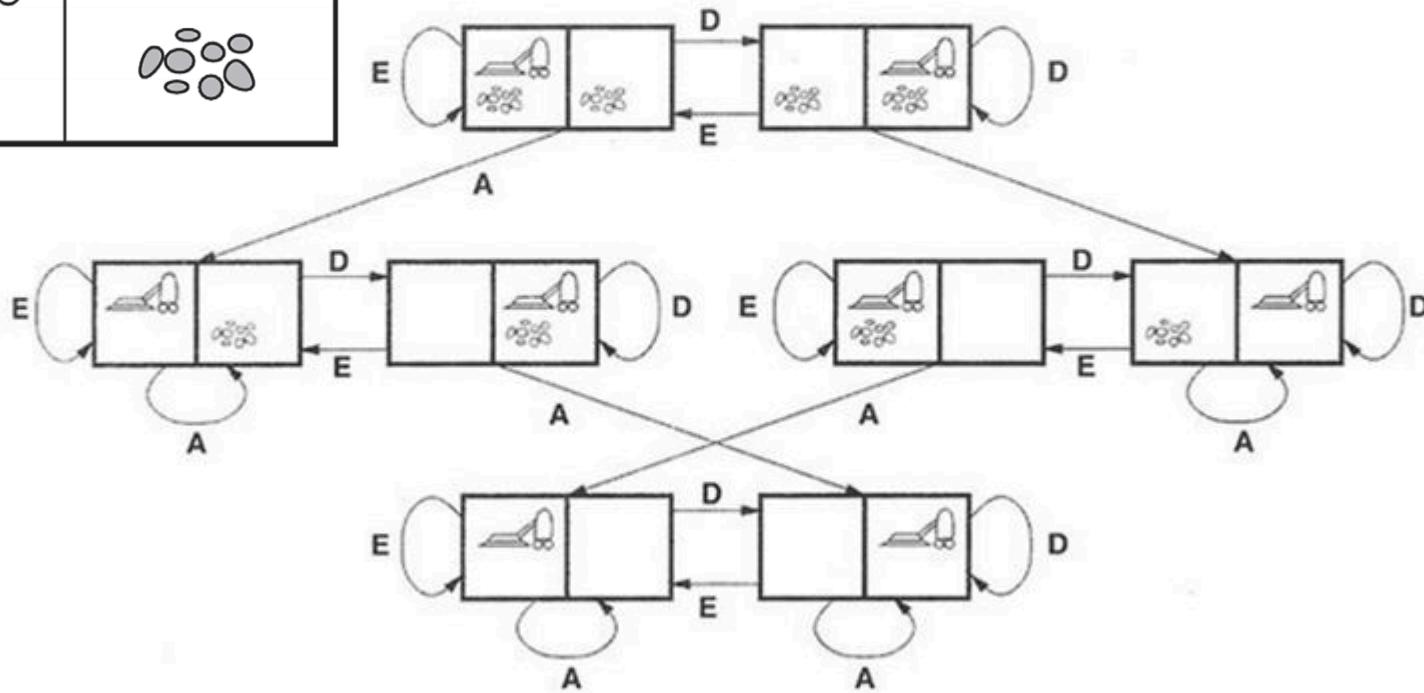
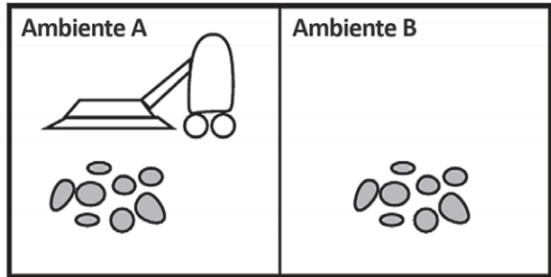
Teste de objetivo: estou em Bucareste?

Função custo: quanto gastei para chegar onde estou?

Problema: CHEGAR EM BUCARESTE
Resultado da busca: o CAMINHO

BUSCAS – Outros exemplos

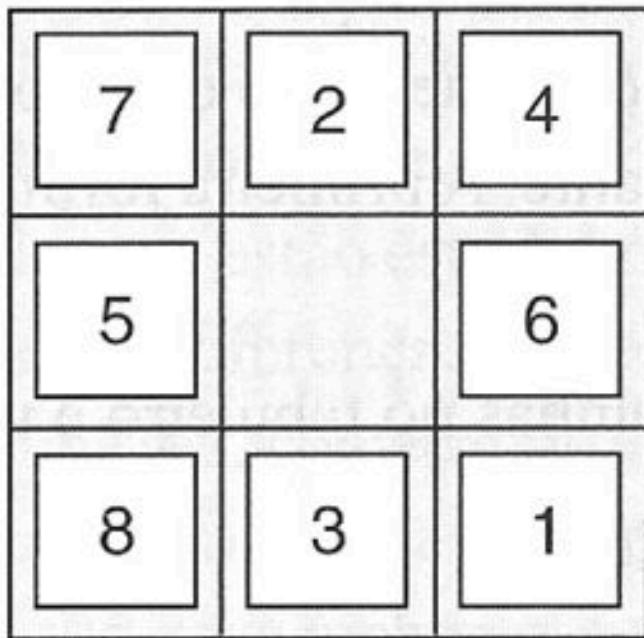
FIAP



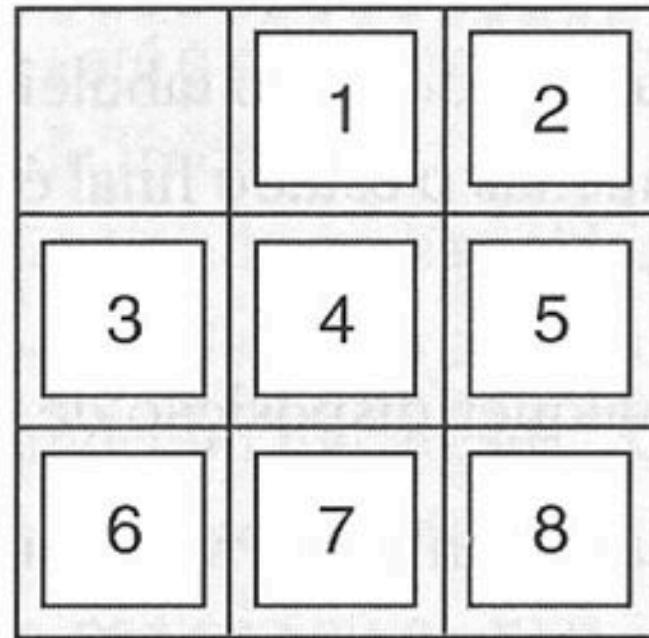
Os arcos denotam ações: E = Esquerda, D = Direita, A = Aspirar

Problema: ambientes limpos
Resultado da busca: sequência de ações

QUEBRA CABEÇA DE 8



Estado inicial

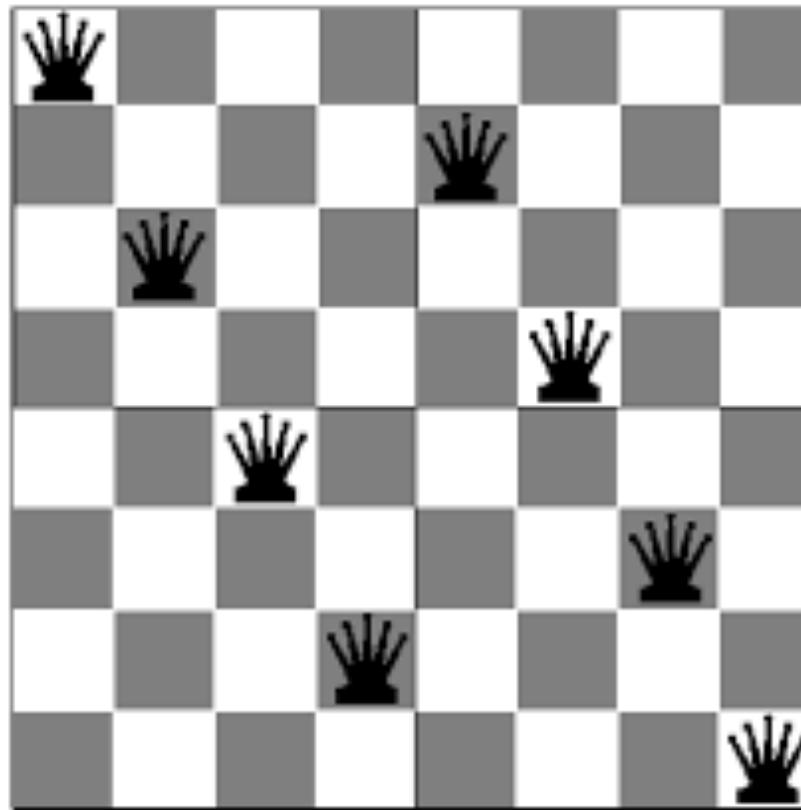


Estado objetivo

Problema: alcançar o estado objetivo

Resultado da busca: sequência de movimentos

BUSCAS – Outros exemplos



Problema: posicionar 8 rainhas no tabuleiro de xadrez de forma que nenhuma rainha possa atacar outra. * Uma rainha ataca na mesma linha, coluna ou diagonal.

Resultado da busca: estado com as rainhas posicionadas como solicitado.

- **Função SUCESSOR (x)**: dado o estado x , retorna um conjunto de pares \langle ação, sucessor \rangle .
 - Cada ação é uma das ações válidas no estado x .
 - Cada sucessor é um estado que pode ser alcançado a partir de x aplicando-se a ação.

SUCESSOR(EM(Arad)) = { \langle IR(Sibiu) ,EM(Sibiu) \rangle ,
 \langle IR(Timissoara) ,EM(Timissoara) \rangle ,
 \langle IR(Zerind) ,EM(Zerind) \rangle }

- **Espaço de estados:** conjunto de todos os estados acessíveis a partir do estado inicial.
 - *Espaço de estados = estado inicial + função SUCESSOR*

EM(Arad) , **EM(Sibiu)** , **EM(Timissoara)** , **EM(Zerind)**

- **Custo do passo:** custo de aplicar a ação a para ir do estado x para o estado y .

CUSTO (<EM(Arad) , EM(Sibiu)>) = 140

- **Busca em árvore:** uma solução tradicional

função BUSCA-EM-ÁRVORE (*problema, estratégia*) retorna uma solução ou uma resposta sobre a “falha”

Iniciar a árvore de busca usando o estado inicial de *problema*
repita

se não existe nenhum candidato para expansão

então retornar falha; // (exit)

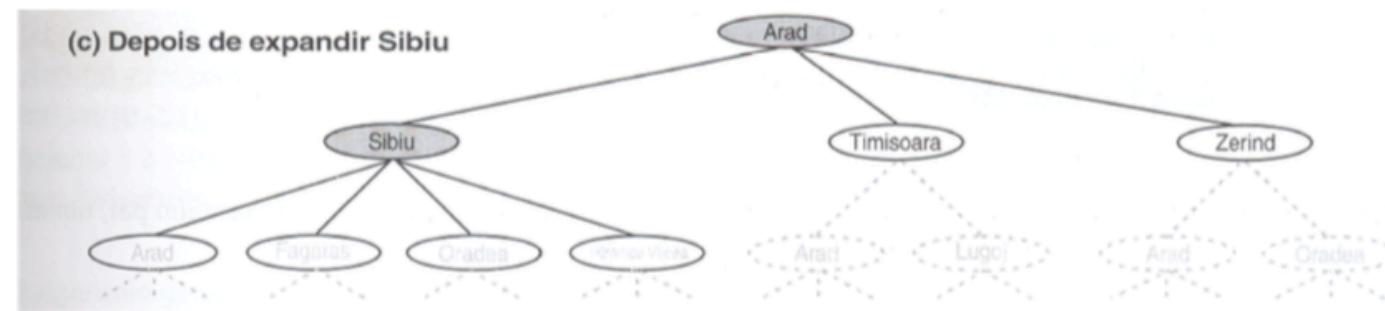
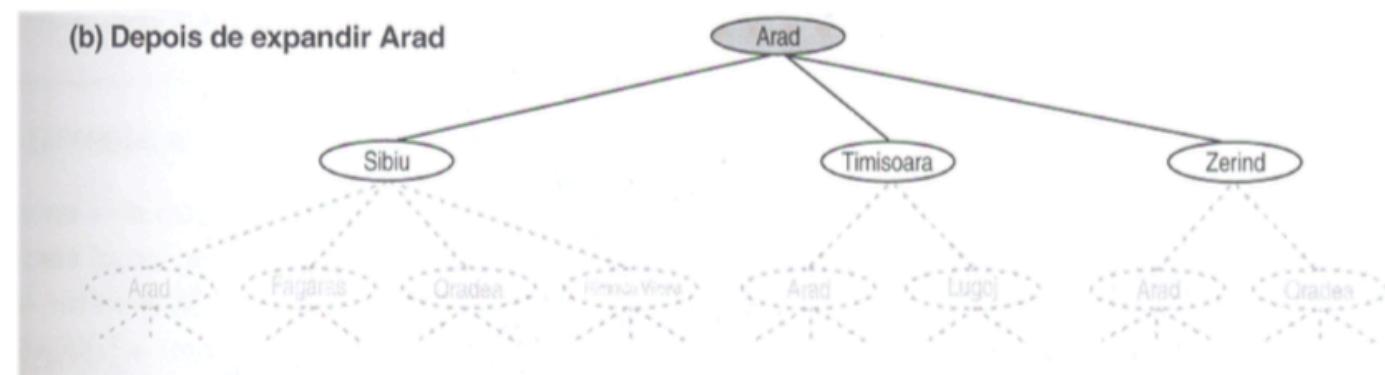
 escolher um nó folha para expansão de acordo com *estratégia*

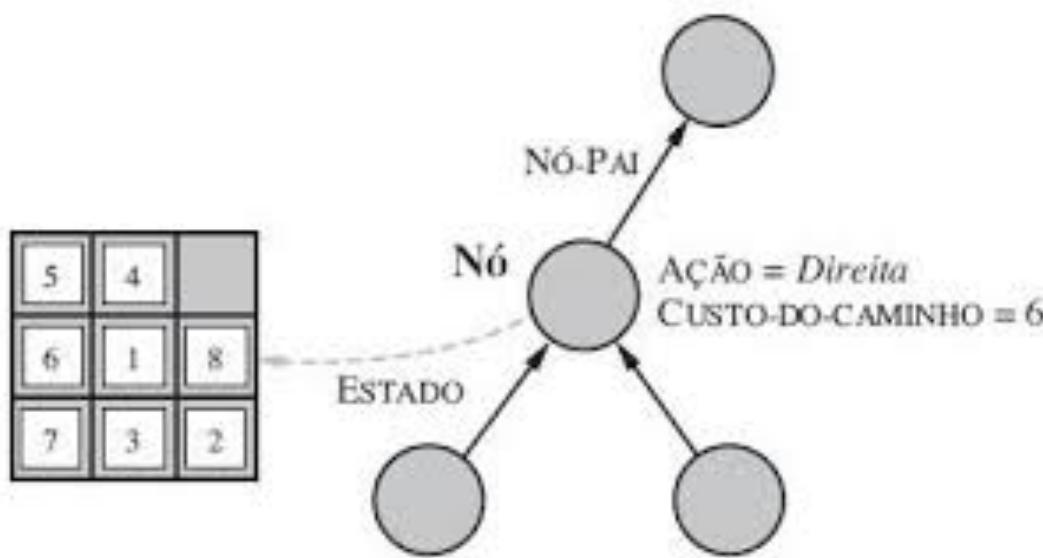
se o nó contém um ***estado objetivo*** então retornar a solução correspondente

senão expandir o nó e adicionar os nós resultantes à árvore de busca

BUSCAS – Árvore: Chegar em Bucareste

FIAP





- **Estado:** o estado a que o nó atual corresponde
- **Nó pai:** nó da árvore de busca que gerou o nó atual
- **Ação:** ação que foi aplicada ao pai para gerar o nó atual
- **Custo do caminho:** custo desde o estado inicial
- **Profundidade:** número de passos ao longo do caminho

Como medir o desempenho do algoritmo usado na resolução do problema?

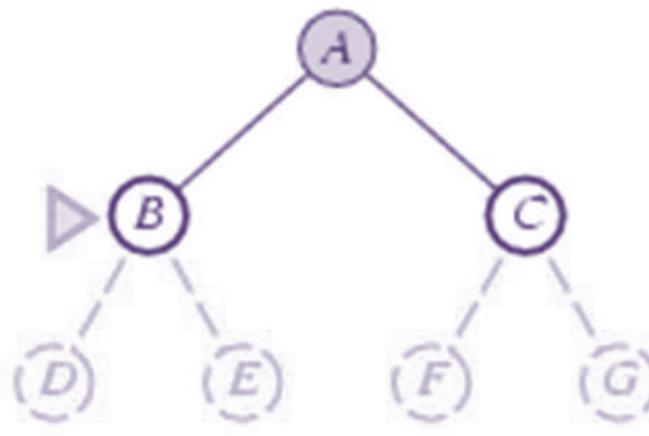
- **Completude:** o algoritmo garante encontrar uma solução quando ela existir?
- **Otimização:** a estratégia encontra a melhor solução? (solução ótima)
- **Complexidade de tempo:** quanto tempo o algoritmo leva para encontrar uma solução?
- **Complexidade de espaço:** quanta memória é necessária para executar a busca?

Como medir o desempenho do algoritmo usado na resolução do problema?

- **Completude:** o algoritmo garante encontrar uma solução quando ela existir?
- **Otimização:** a estratégia encontra a melhor solução? (solução ótima)
- **Complexidade de tempo:** quanto tempo o algoritmo leva para encontrar uma solução?
- **Complexidade de espaço:** quanta memória é necessária para executar a busca?

- **Busca cega ou busca sem informação:** estratégias que não usam informação adicional sobre os estados além das fornecidas na definição do problema – só gera sucessores e distingue um estado objetivo dos outros.
 - Busca em extensão
 - Busca de custo uniforme
 - Busca em profundidade
 - Busca em profundidade limitada
 - Busca de aprofundamento iterativo
- **Busca com informação e heurística:** estratégias que sabem se um estado não-objetivo é mais promissor do que outro.

- **Busca em extensão:** a cada passo, são expandidos os nós para seus sucessores.



- **Busca de custo uniforme:** expande-se o nó n com o caminho de custo mais baixo.
 - Se todos os “custos de passos” forem iguais, essa busca é igual à busca em extensão.

- **Busca em profundidade:** expande o nó mais profundo na borda da árvore que ainda não foi expandido.



- **Busca em profundidade limitada:** nós na profundidade p são tratados como se não tivessem sucessores.
 - Resolve o problema de caminhos infinitos, mas pode não chegar na solução.
- **Busca em profundidade, com aprofundamento interativo:** aumenta gradualmente o limite, até alcançar o objetivo.

- **Busca com informação:** usa conhecimento específico sobre o problema, além da definição do problema.
- **Busca heurística ou busca pela melhor escolha:** a seleção do nó a ser expandido é feita com base em uma avaliação do nó, aplicando uma função de avaliação $f(n)$.

A **função de avaliação**, em geral, mede de alguma forma a **distância** do nó n até o objetivo (por exemplo, usando algum tipo de custo). Assim, o **nó selecionado** é o com avaliação de **valor mais baixo**.

Cuidado “busca pela melhor escolha” não é uma marcha direta ao objetivo, é a escolha do nó que PARECE ser o melhor de acordo com a função de avaliação!

- **Busca gulosa:** guiada pela melhor escolha, expande o nó mais próximo à meta, supondo que isso provavelmente levará a uma solução rápida e eficiente.

Avalia os nós usando apenas a **função heurística**: $f(n) = h(n)$.

Função heurística $h(n)$:

- Custo **estimado** do caminho mais econômico do nó n até um nó objetivo.
- Avaliação do próprio nó como uma solução para o problema.
- Função arbitrária específica para um problema, com uma restrição:
Para um nó objetivo, $h(n) = 0$.

BUSCAS – Com informação

Voltando ao problema de chegar a Bucareste...

Heurística: distância em linha reta até Bucareste – H_{DLR}

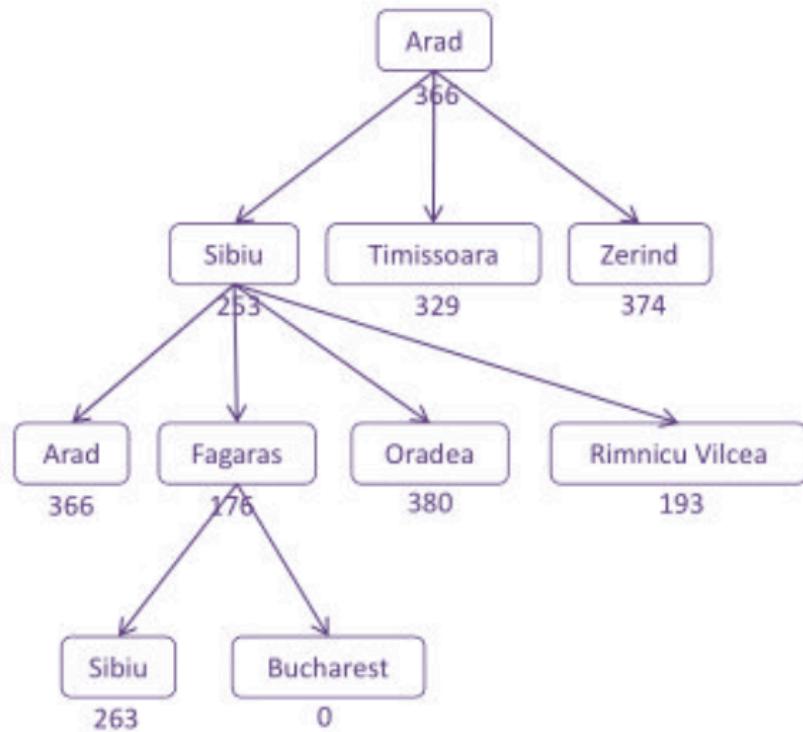
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374
Hirsova	151	Urziceni	80

Os valores H_{DLR} não fazem parte dos dados na definição do problema!

É necessário experiência no **contexto** do problema para saber que H_{DLR} é uma heurística útil por estar relacionada com distâncias reais.

Voltando ao problema de chegar a Bucareste...

Heurística: distância em linha reta até Bucareste – H_{DLR}



Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374
Hirsova	151	Urziceni	80

A heurística não é ótima: o Caminho passando por Sibiu e Fagaras é 32 quilômetros mais longo que o caminho por Rimmicu Vilcea e Pitesti.

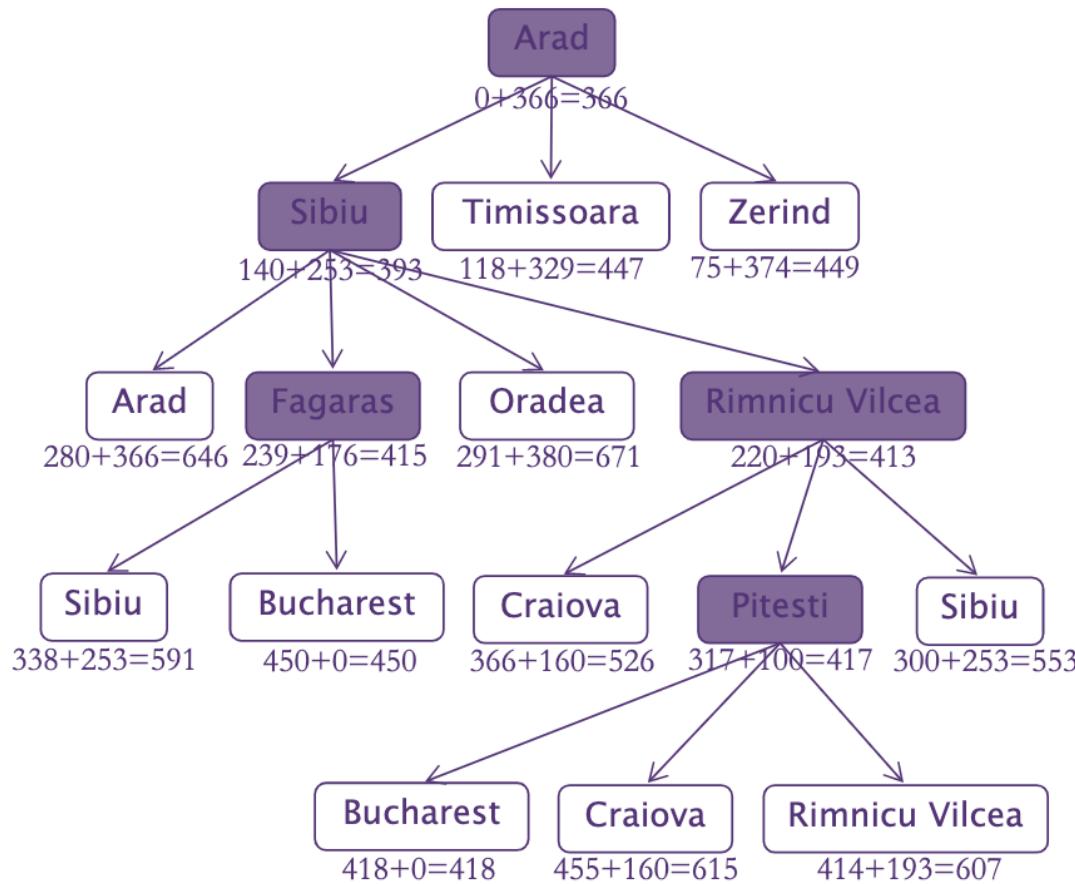
- **Busca A***: também guiada pela melhor escolha, mas minimiza o custo total estimado da solução.
 - Função de avaliação $f(n) = g(n) + h(n)$.
 - $g(n)$ é o custo para alcançar cada nó e $h(n)$ é o custo para ir do nó atual até o objetivo.
 - $h(n)$ deve ser admissível: nunca superestimar o custo para alcançar o objetivo.
 - Permite voltar uma decisão.

BUSCAS – Com informação

FIAP

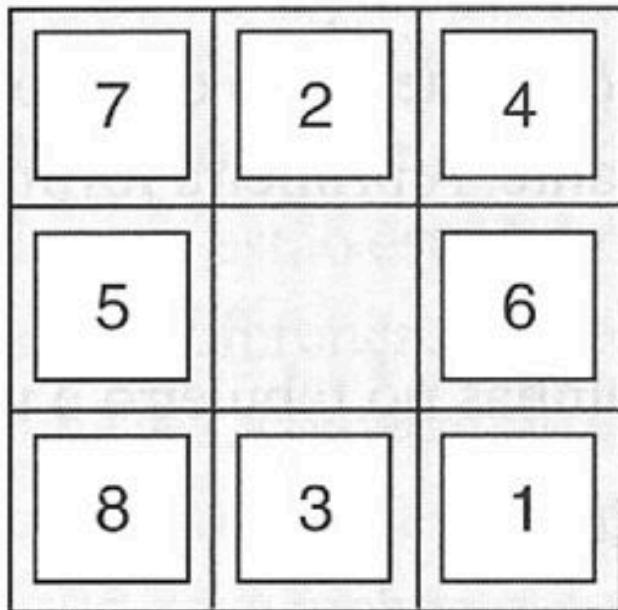
Voltando ao problema de chegar a Bucareste...

Heurística: distância em linha reta até Bucareste – H_{DLR}

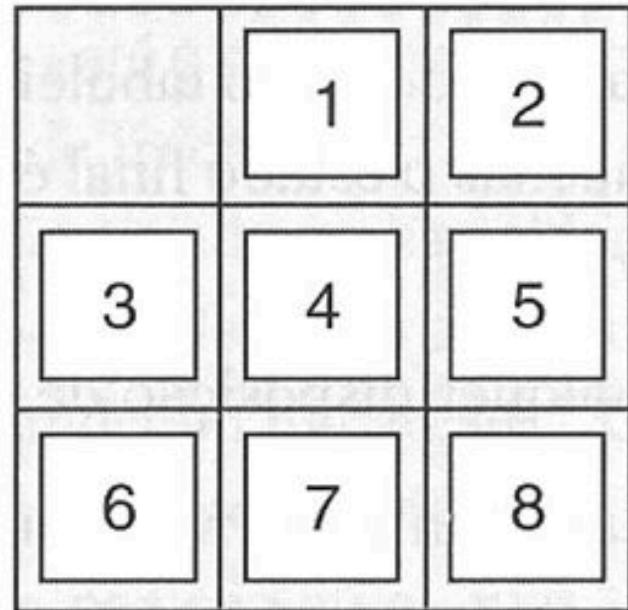


BUSCAS – Com informação

Para o problema do **quebra-cabeças de 8 peças**, qual poderia ser a **heurística** adotada?



Estado inicial



Estado objetivo

Para usarmos a busca A*, precisamos definir uma função heurística que **nunca superestime o número de passos até o objetivo**.

Copyright © 2019 Profa. Thais Rodrigues Neubauer

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).