Redes Neurais (Inteligência Artificial)

Aula 04 – Busca Heurística

Edirlei Soares de Lima
<edirlei@iprj.uerj.br>

Métodos de Busca

- **Busca Cega ou Exaustiva:**
  - Não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**
  - **Estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas.**

- **Busca Local:**
  - Operam em um único estado e movem-se para a vizinhança deste estado.

Busca Heurística

- **Algoritmos de Busca Heurística:**
  - Busca Gulosa
  - A*

- A busca heurística leva em conta o **objetivo** para decidir qual caminho escolher.

- Conhecimento extra sobre o problema é utilizado para **guiar o processo de busca**.

Busca Heurística

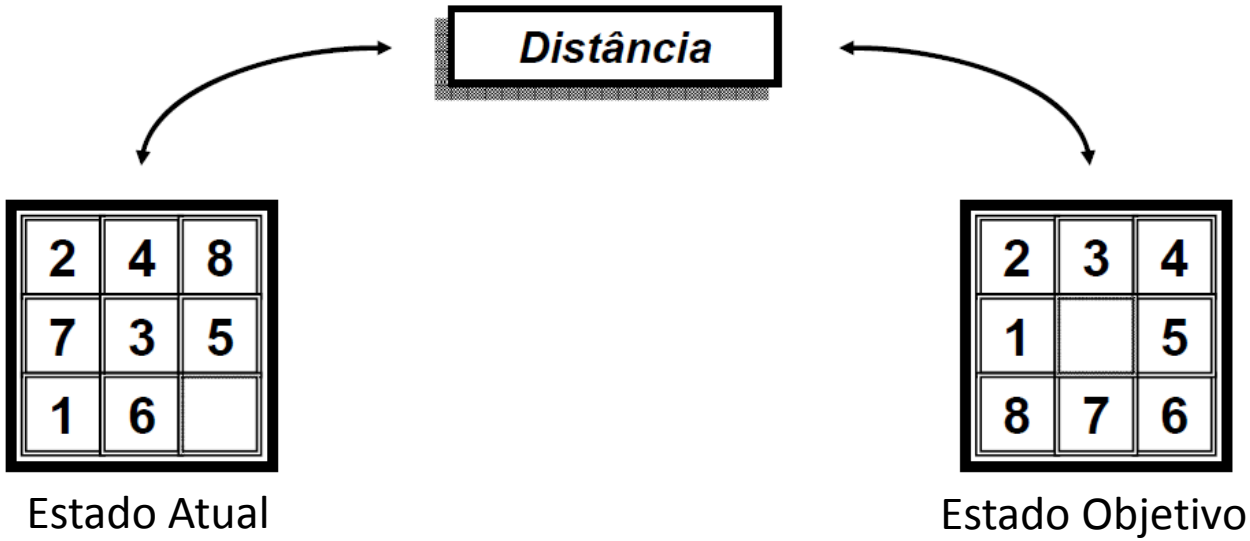
- Como encontrar um barco perdido?
  - **Busca Cega** -> Procura no oceano inteiro.
  - **Busca Heurística** -> Procura utilizando informações relativas ao problema.
    - Exemplo: correntes marítimas, vento, etc.

Busca Heurística

- **Função Heurística** (*h*)
  - Estima o custo do caminho mais barato do estado atual até o estado final mais próximo.
  - São específicas para cada problema.

- **Exemplo:**
  - Encontrar a rota mais curta entre duas cidades:
    - *h*(*n*) = distância em linha reta direta entre o nó *n* e o nó final.

Função Heurística



Estado Atual

Estado Objetivo

$$h_1\left(\begin{array}{|c|c|c|}\hline 2&4&8\\\hline 7&3&5\\\hline 1&6&\\\hline\end{array}\right) = ?$$

Busca Heurística

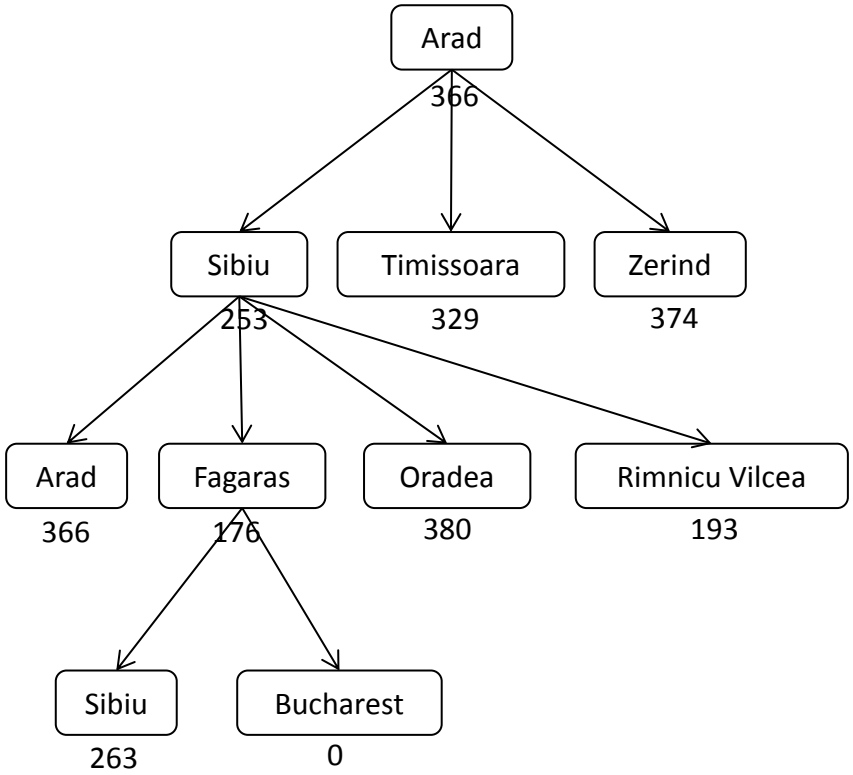
- **Algoritmos de Busca Heurística:**
  - Busca Gulosa
  - A*

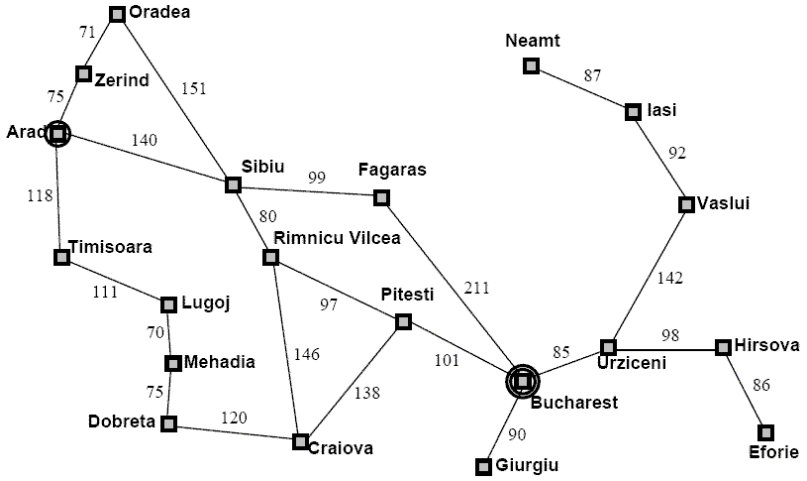
Busca Gulosa

- **Estratégia:**
  - Expande os nós que se encontram mais próximos do objetivo (uma linha reta conectando os dois pontos no caso de distancias), desta maneira é provável que a busca encontre uma solução rapidamente.

- A implementação do algoritmo se assemelha ao utilizado na busca cega, entretanto utiliza-se uma função heurística para decidir qual o nó deve ser expandido.

Busca Gulosa





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

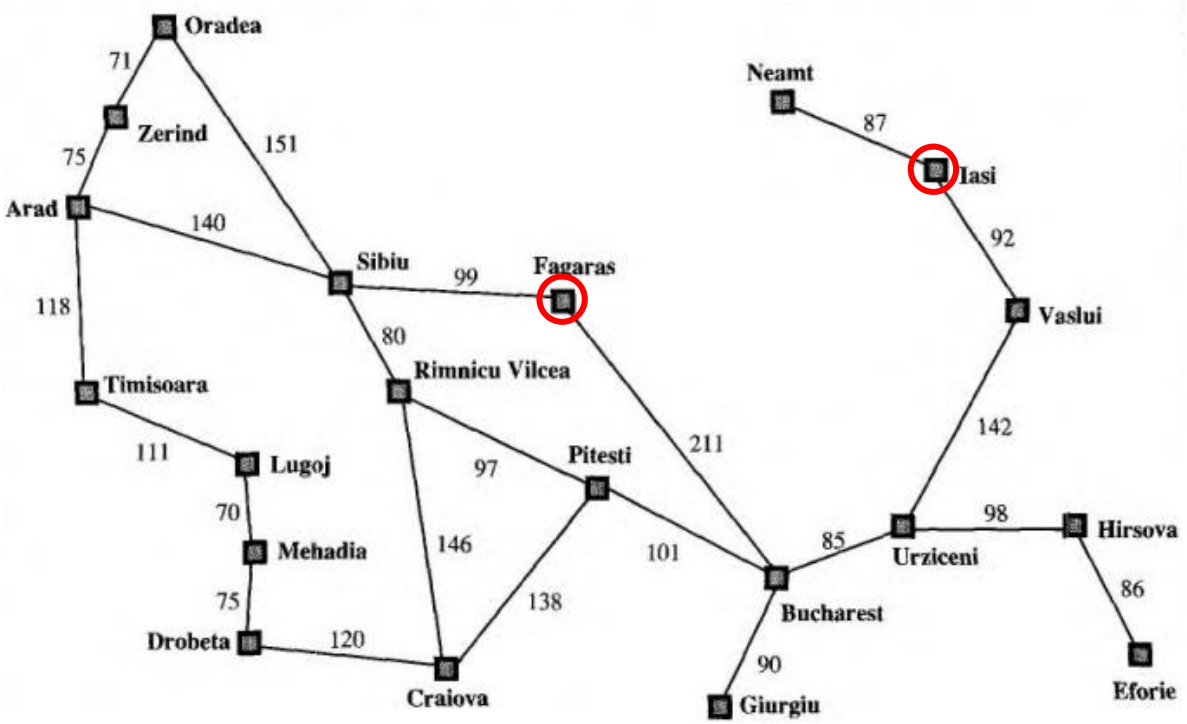
Função Heurística (h):
Distancia em linha reta

Busca Gulosa

- **Custo de busca mínimo:**
  - No exemplo, não expande nós fora do caminho.
- **Não é ótima:**
  - No exemplo, escolhe o caminho que é mais econômico à primeira vista, via Fagaras.
  - Porém, existe um caminho mais curto via Rimnicu Vilcea.
- **Não é completa:**
  - Pode entrar em loop se não detectar a expansão de estados repetidos.
  - Pode tentar desenvolver um caminho infinito.

Busca Gulosa

- Ir de **Iasi** para **Fagaras**?

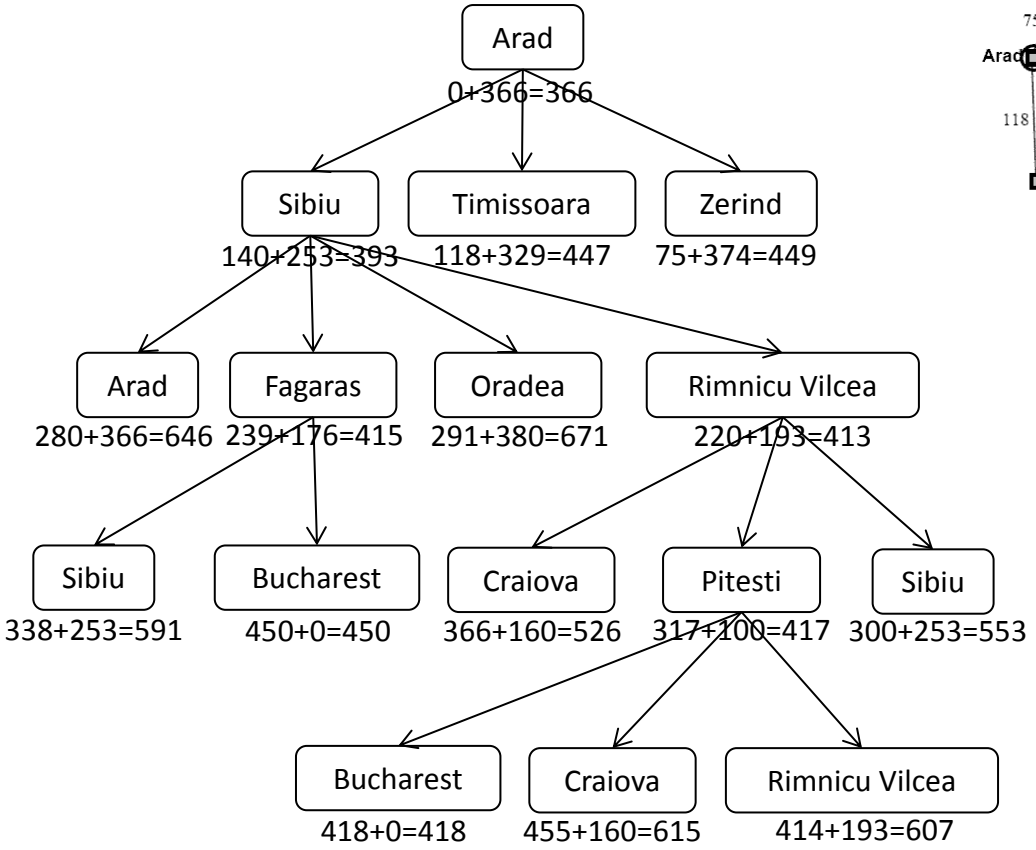


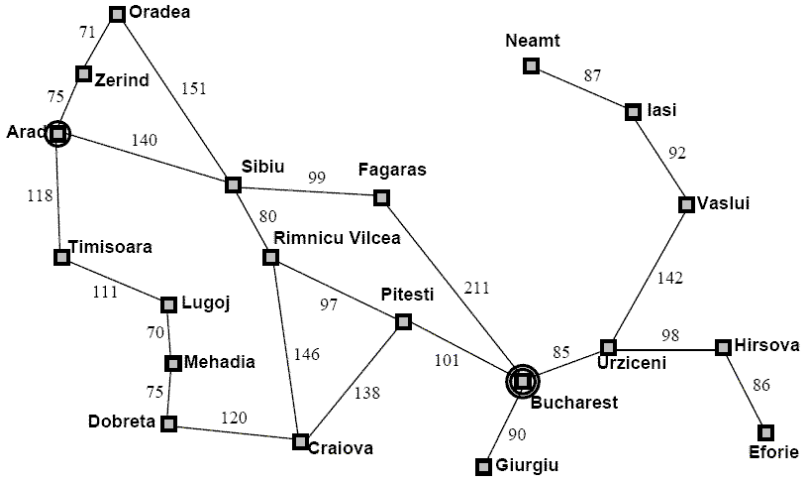
Busca A*

- **Estratégia:**
  - Combina o custo do caminho $g(n)$ com o valor da heurística $h(n)$
  - $g(n)$ = custo do caminho do nó inicial até o nó n
  - $h(n)$ = valor da heurística do nó n até um nó objetivo (distancia em linha reta no caso de distancias espaciais)
  - $f(n) = g(n) + h(n)$

- **É a técnica de busca mais utilizada.**

Busca A*





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

Busca A*

- A estratégia é **completa** e **ótima**.

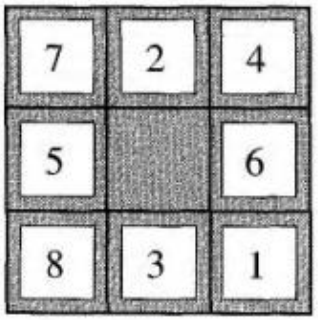
- **Custo de tempo:**
  - Exponencial com o comprimento da solução, porém boas funções heurísticas diminuem significativamente esse custo.

- **Custo memória:**
  - Guarda todos os nós expandidos na memória.

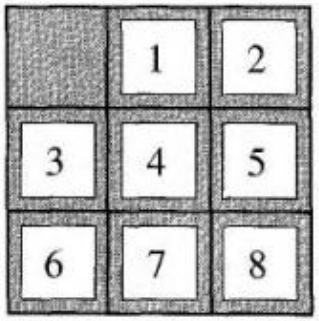
- Nenhum outro algoritmo ótimo garante expandir menos nós.

Definindo Heurísticas

- Cada problema **exige** uma função heurística diferente.
- Não se deve superestimar o custo real da solução.
- Como escolher uma boa função heurística para o jogo 8-Puzzle?

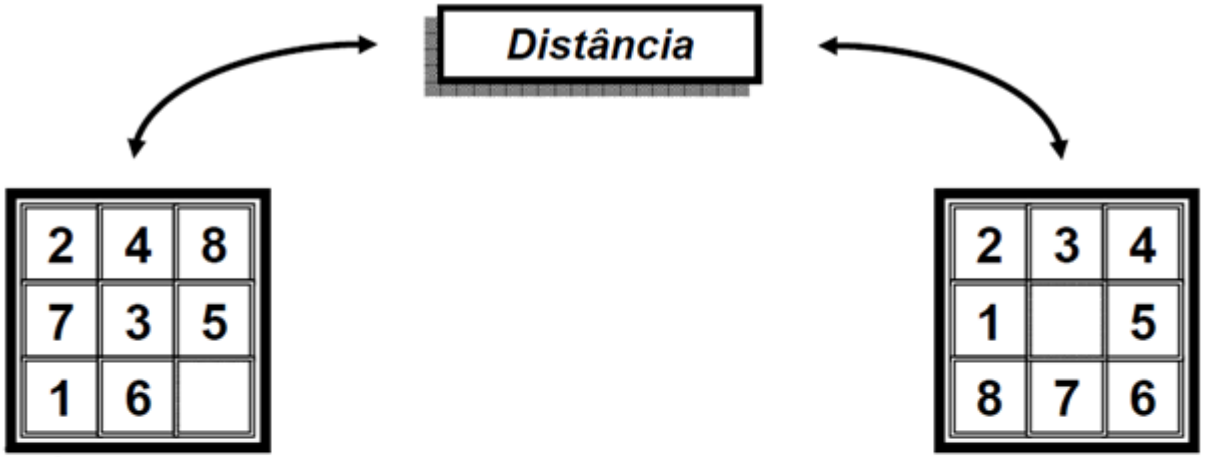


Start State



Goal State

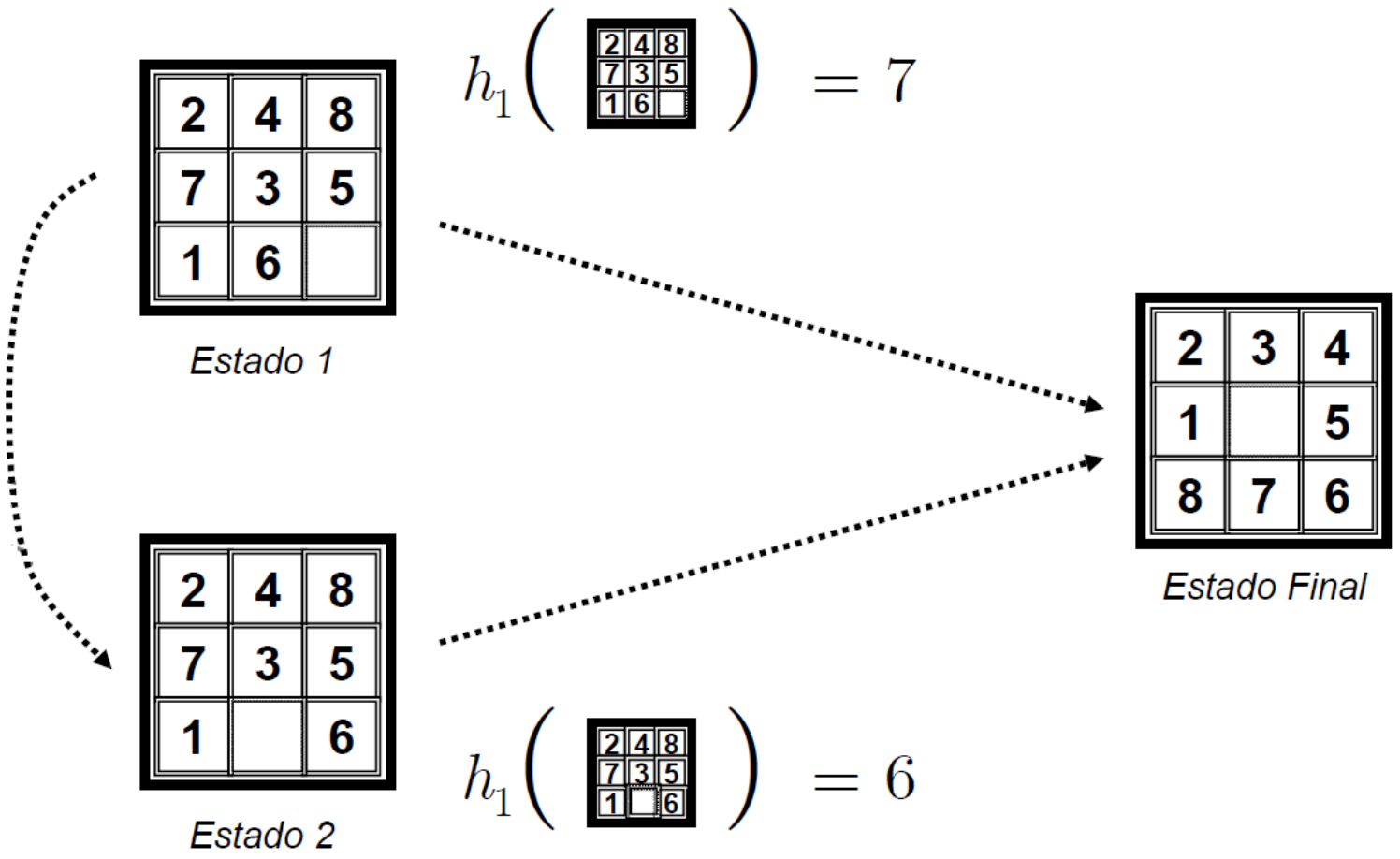
Definindo Heurísticas



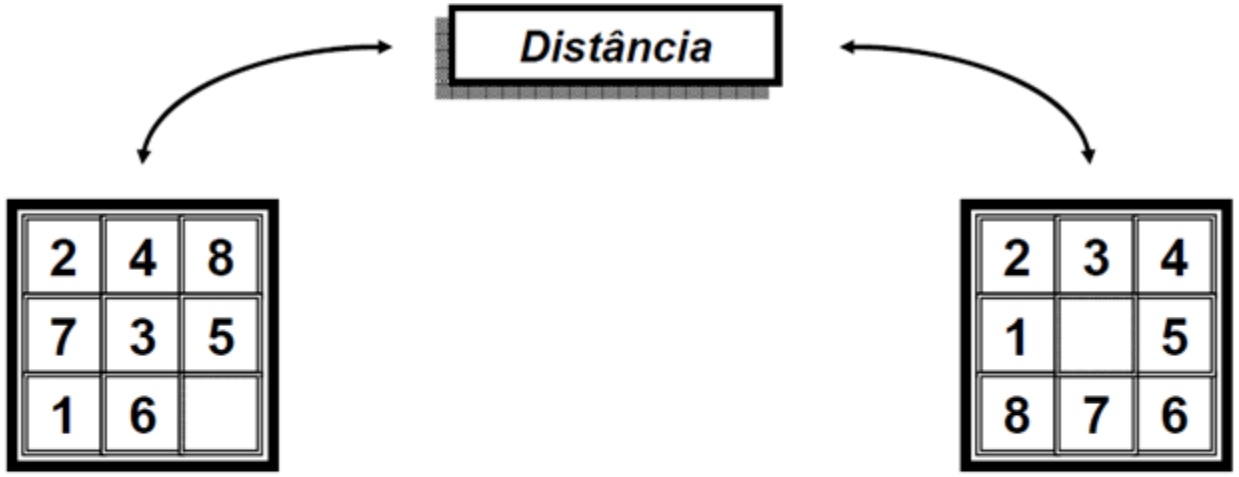
$$h_1\left(\begin{array}{|c|c|c|}\hline 2&4&8\\\hline 7&3&5\\\hline 1&6&\\\hline\end{array}\right) = \textit{A quantidade de peças for a do lugar}$$

$$= 7$$

Definindo Heurísticas



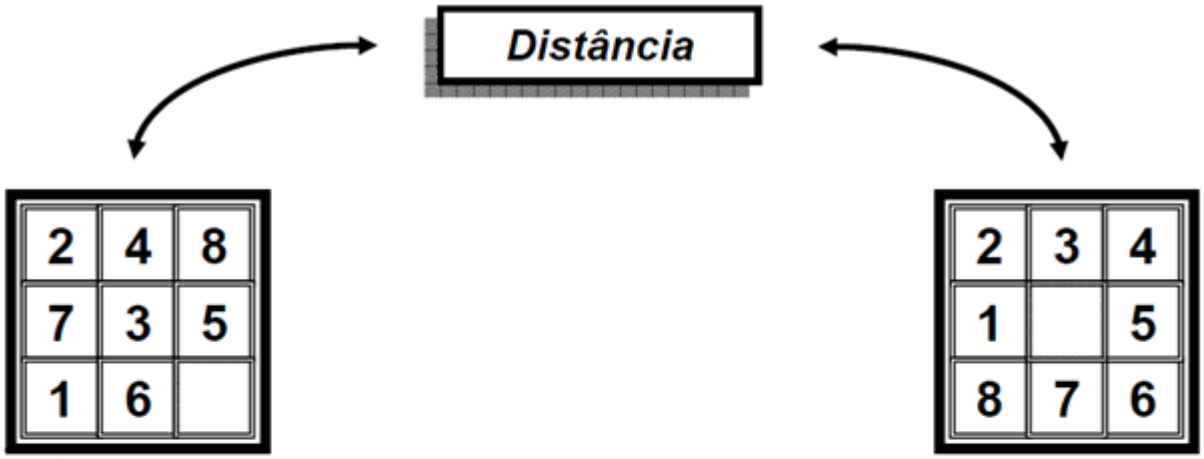
Definindo Heurísticas



$$h_2\left(\begin{array}{|c|c|c|}\hline 2&4&8\\\hline 7&3&5\\\hline 1&6&\\\hline\end{array}\right) = ?$$

Outra Heurística?

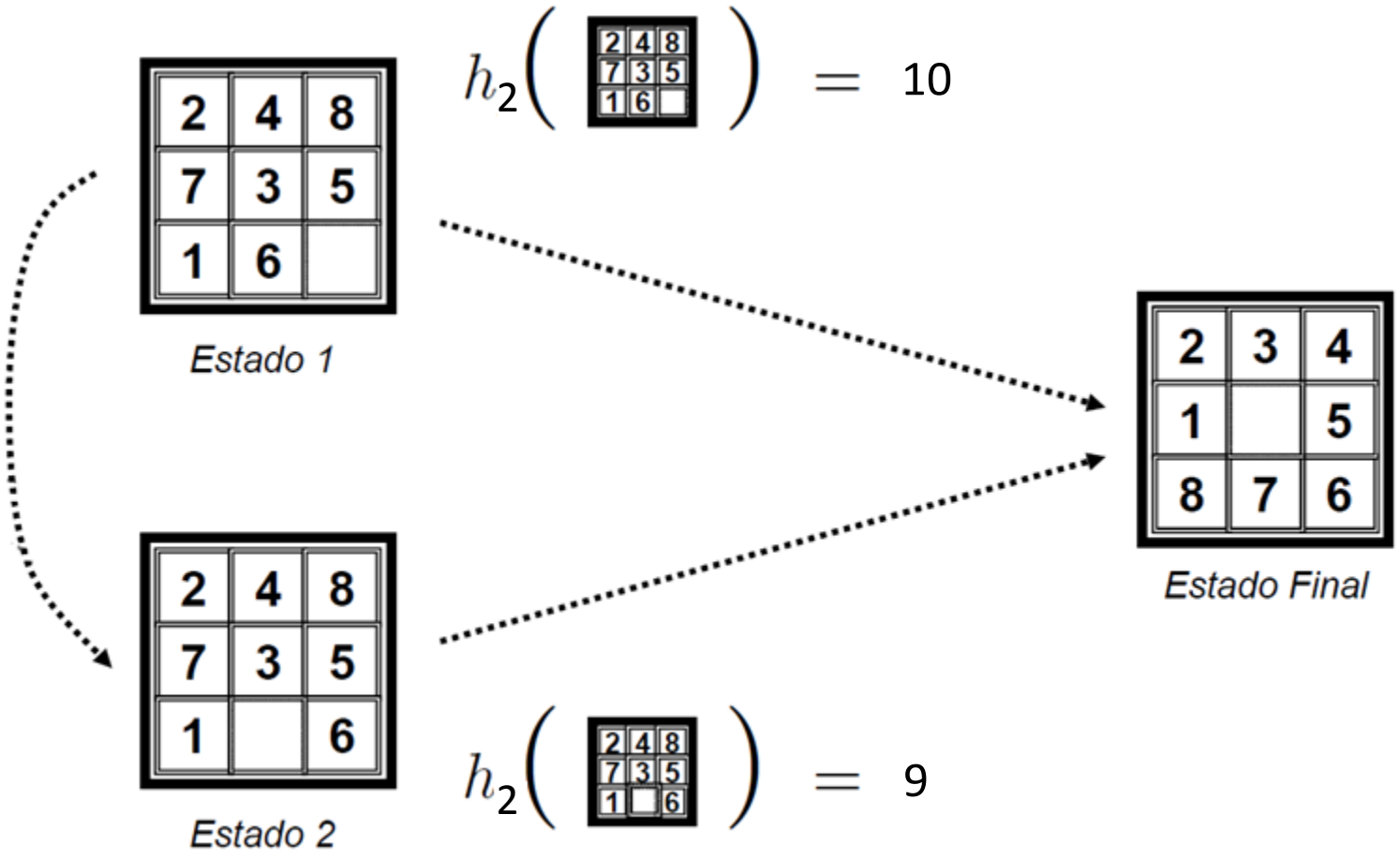
Definindo Heurísticas



$$h_2\left(\begin{array}{|c|c|c|}\hline 2&4&8\\\hline 7&3&5\\\hline 1&6&\\\hline\end{array}\right) = \textit{Número de movimentos necessários para colocar cada peça no seu lugar}$$

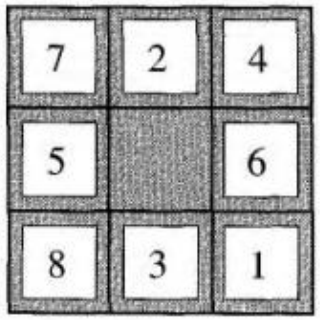
$$= 10$$

Definindo Heurísticas

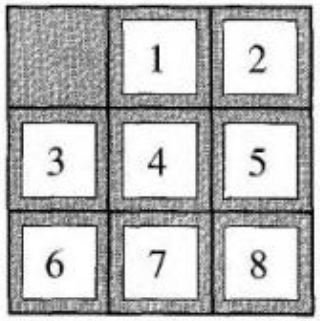


Definindo Heurísticas

- Como escolher uma boa função heurística para o jogo 8-Puzzle?
  - h^1 = número de elementos fora do lugar.
  - h^2 = soma das distâncias de cada número à sua posição final (movimentação horizontal e vertical).
- Qual das heurísticas é melhor?

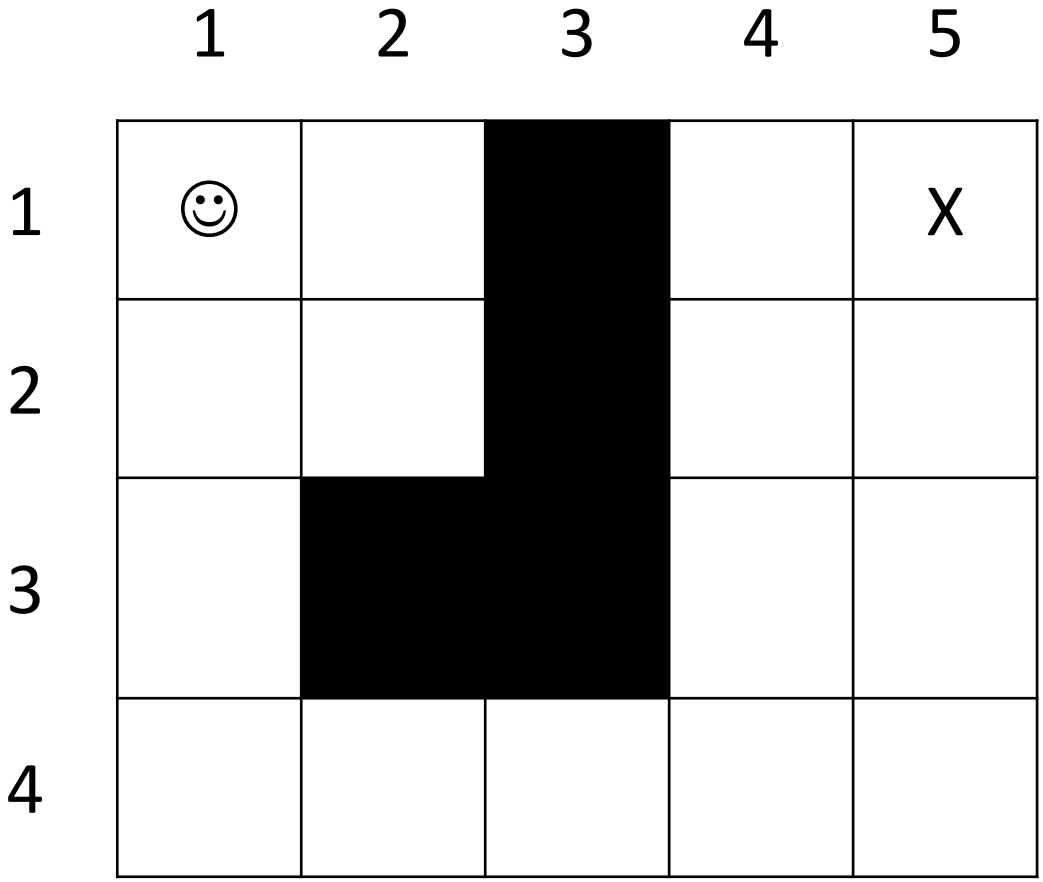


Start State



Goal State

Exemplo - A*



Exemplo - A*

- Qual é o espaço de estados?
- Quais são as ações possíveis?
- Qual será o custo das ações?

Exemplo - A*

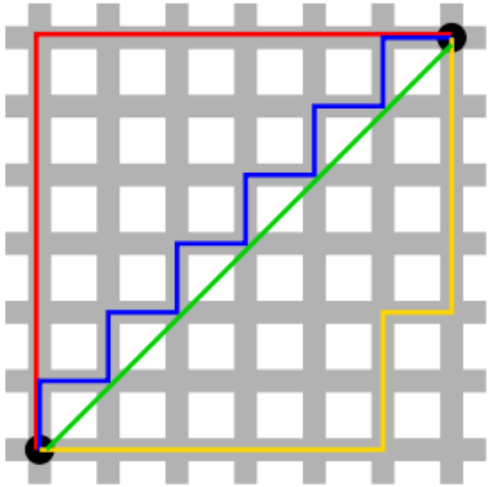
- **Heurística do A*:** $f(n) = g(n) + h(n)$
  - $g(n)$ = custo do caminho
  - $h(n)$ = função heurística

- Qual seria a função heurística $h(n)$ mais adequada para este problema?

  - A distancia em linha reta é uma opção.

Exemplo - A*

- Como calcular a heurística *h*(*n*)?
  - Distancia de Manhattan



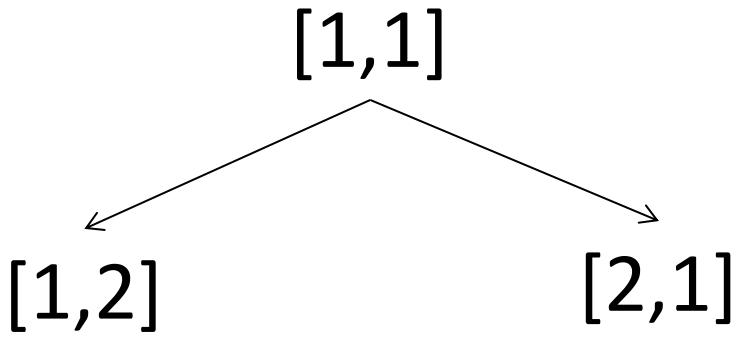
$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

Exemplo - A*

- O próximo passo é gerar a árvore de busca e expandir os nós que tiverem o menor valor resultante da função heurística $f(n)$.

  - $f(n) = g(n) + h(n)$

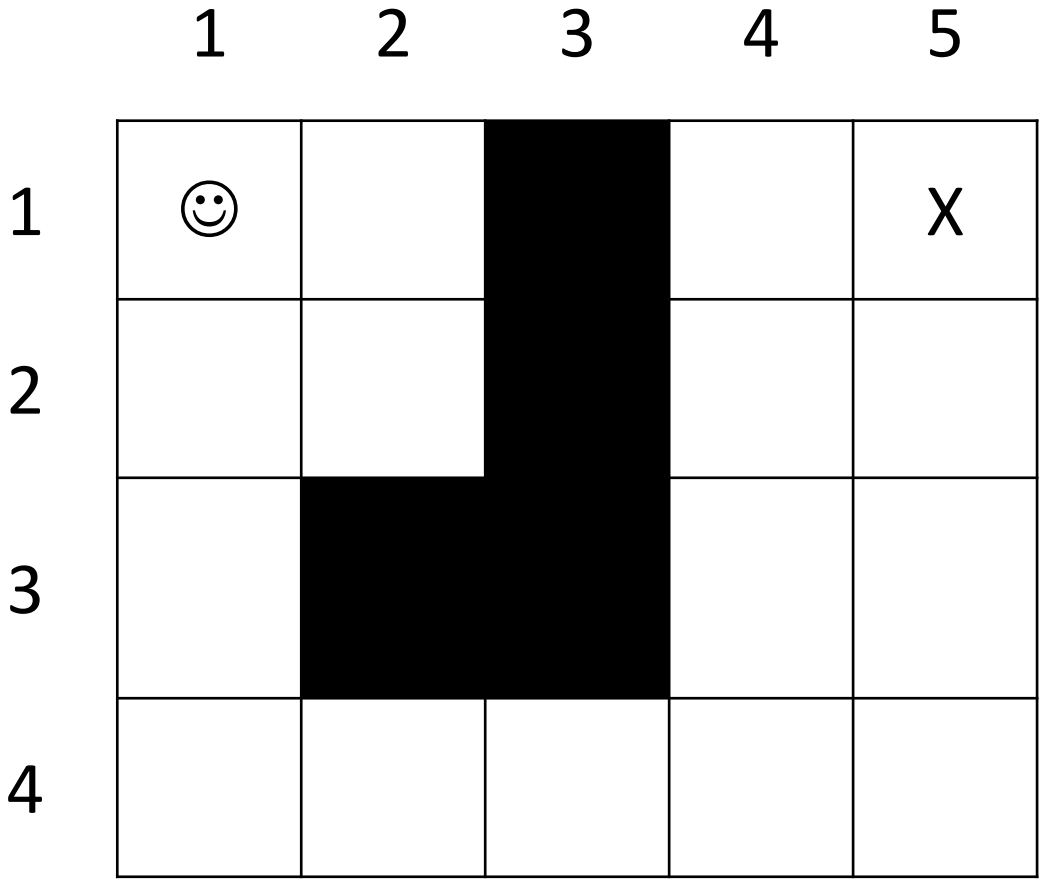
Exemplo - A*



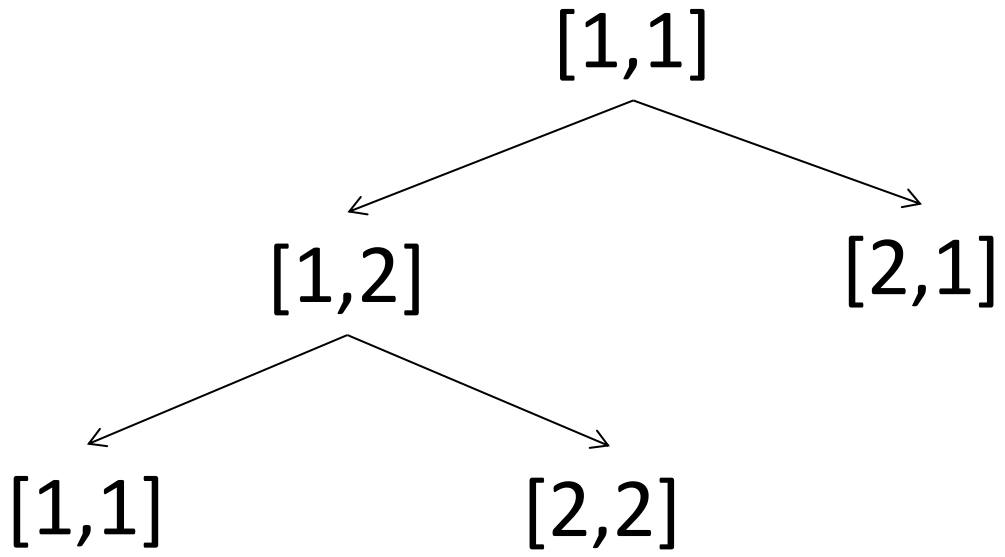
[1,2] = $f(n)$ = ?? + ??

[2,1] = $f(n)$ = ?? + ??

Exemplo - A*



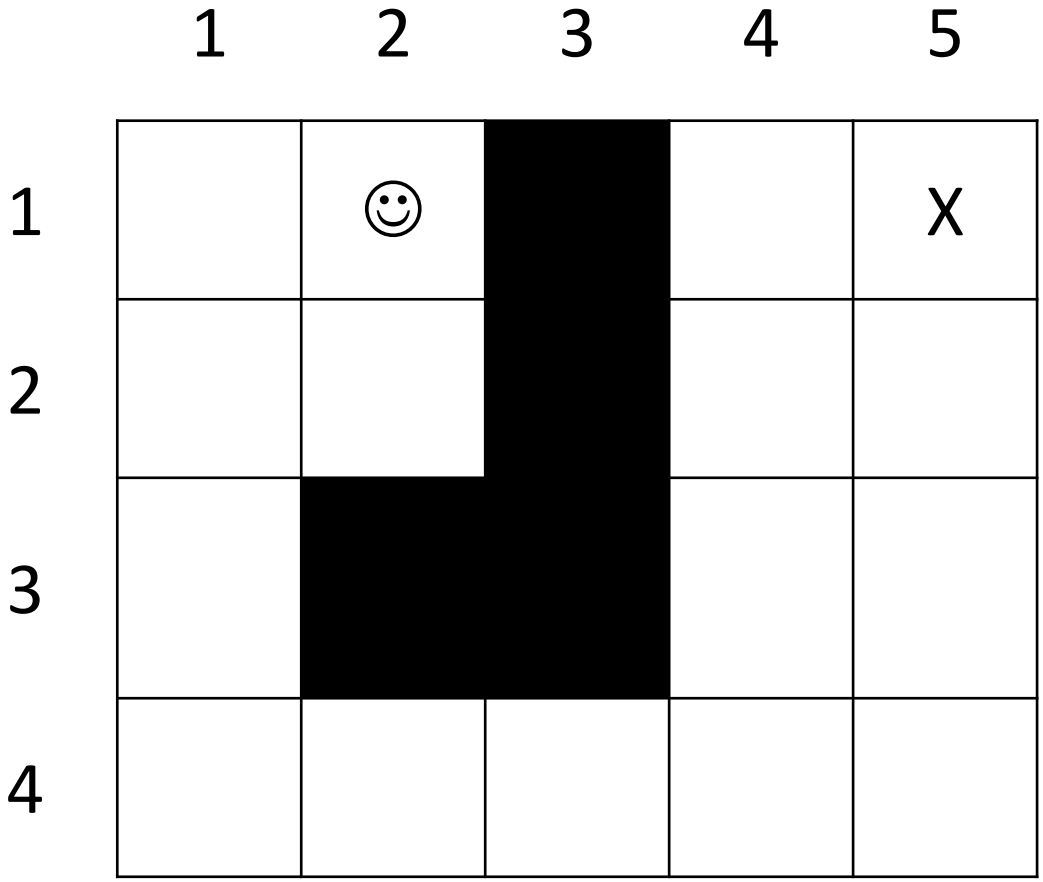
Exemplo - A*



[1,1] = $f(n)$ = ?? + ??

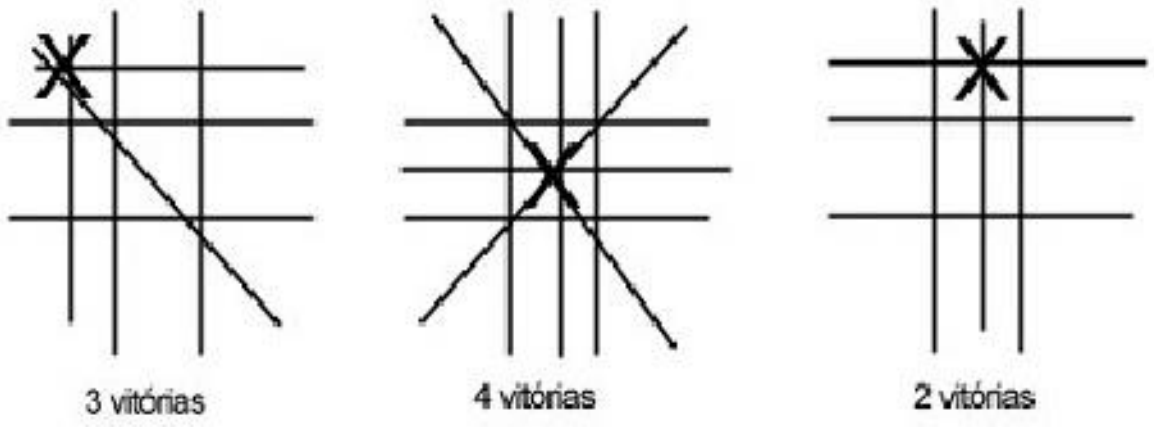
[2,2] = $f(n)$ = ?? + ??

Exemplo - A*



Exercícios

- (1) Qual seria uma boa heurística para o **jogo da velha**?



Exercícios

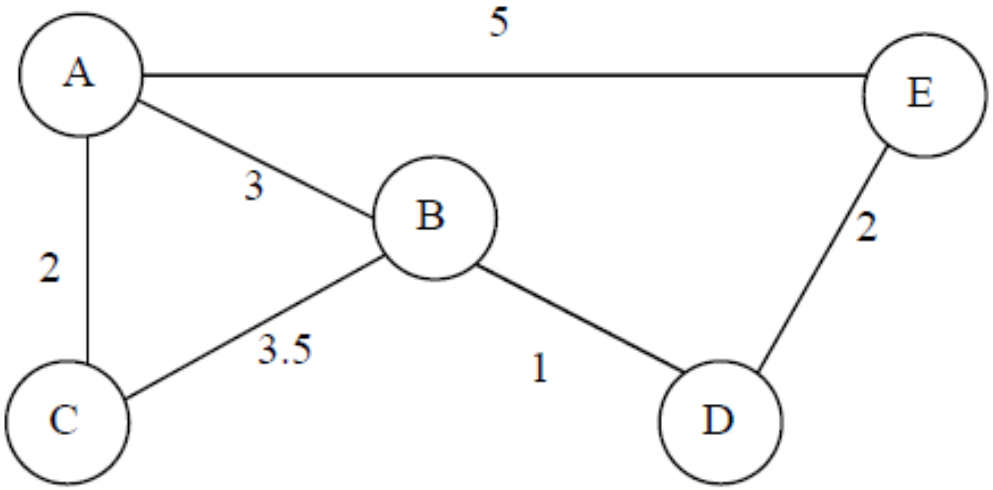
- (2) Supondo que é necessário utilizar um algoritmo de busca para resolver um problema no qual são necessárias **respostas instantâneas**. Mas, mesmo utilizando o A* com uma boa função heurística, o tempo gasto com o processo de busca ainda está muito grande. O que pode ser feito para otimizar esse processo?

  - Caminhos pré-calculados.
  - Custos pré-calculados.

Caminhos Pré-Calculados

- **Tabela pré-calculada** com os melhores caminhos.

- Armazena-se somente o próximo nó que deve ser seguindo do nó atual ao nó destino.



	A	B	C	D	E
A	A	B	C	B	E
B	A	B	C	D	D
C	A	B	C	B	**B**
D	B	B	B	D	E
E	A	D	D	D	E

Custos Pré-Calculados

- Saber qual o melhor caminho entre dois nós somente é útil quando se sabe onde se deseja ir.

- Uma **tabela pré-calculada** com os **custos de locomoção** entre quaisquer dois nós também é uma informação muito util.

	A	B	C	D	E
A	0	3	2	4	5
B	3	0	3.5	1	3
C	2	3.5	0	4.5	6.5
D	4	1	4.5	0	2
E	5	3	6.5	2	0

Leitura Complementar

- Russell, S. and Norvig, P. **Artificial Intelligence: a Modern Approach**, 3nd Edition, Prentice-Hall, 2009.

- **Capítulo 4: Informed Search and Exploration**