



# INF 1771 – Inteligência Artificial

Aula 04 – Busca Heurística

# 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 em largura;
  - Busca em profundidade;
  - Busca de custo uniforme;
  - Busca com aprofundamento iterativo;

- **Busca com Informação ou 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 com Informação

- Usa conhecimento específico sobre o problema para encontrar soluções de forma mais eficiente do que a busca cega.
- Pode ser mais eficiente do que a busca cega.

# 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

- O conhecimento do problema é codificado em uma função **heurística:  $h(n)$** 
  - computada para cada estado
  - estima
    - o quão bom um nó é
    - o quão perto um nó está de chegar na solução

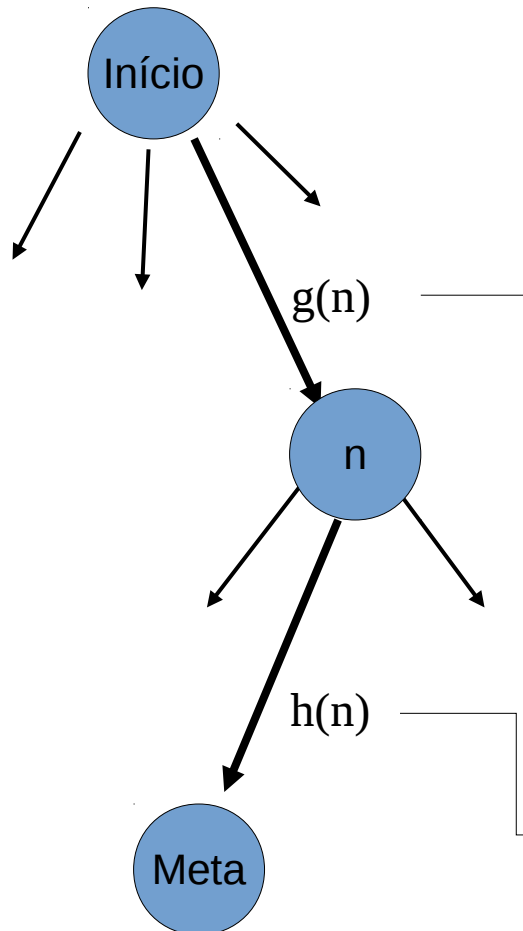
# Busca Heurística

- $h(n)$  é maior ou igual a zero para todos os nós
- $h(n) = 0$ : objetivo

# Busca Heurística: função de avaliação

- A função heurística irá compor uma função de avaliação:  $f(n)$
- $f(n)$  quantifica a qualidade de uma solução.
- Geralmente é composta pela:
  - Função de custo  $g(n)$
  - Função heurística  $h(n)$

# Busca Heurística: função de avaliação



Podemos usar o custo do caminho  $g(n)$  para decidir que nó expandir como na Busca de Custo Uniforme. Mas esta medida não dirige a busca para o nó objetivo, ela cuida do **passado**.

A função heurística  $h(n)$  calcula o custo estimado para se alcançar o nó objetivo. Ela cuida do **futuro**.



# Busca Heurística:

## Estratégia

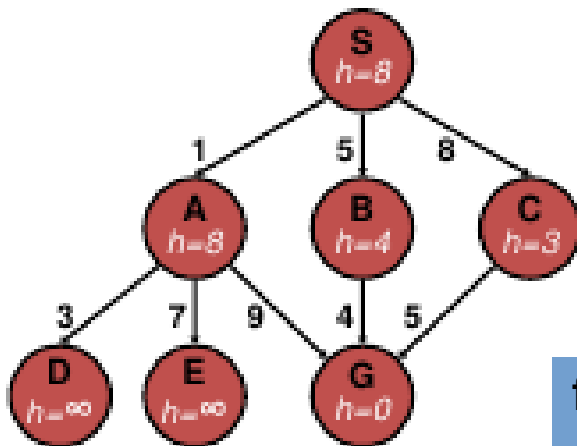
- Estratégia geral

- Busca pela melhor escolha (estimativa) – Best First
- Expandir nó mais desejável que ainda não foi expandido

# Busca gulosa

- Busca que usa somente a heurística como função de avaliação
  - $f(n) = h(n)$
- Expande o nó que parece estar mais próximo da meta, seguindo  $h(n)$

# Busca Gulosa: exemplo

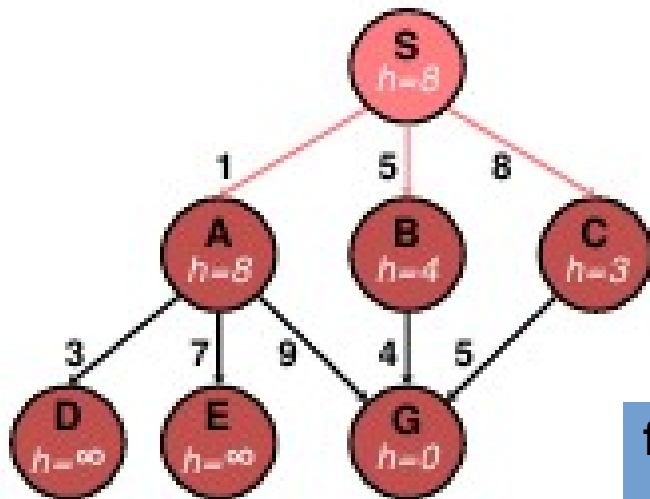


frontier

expanded

[S:8]

# Busca Gulosa: exemplo



frontier

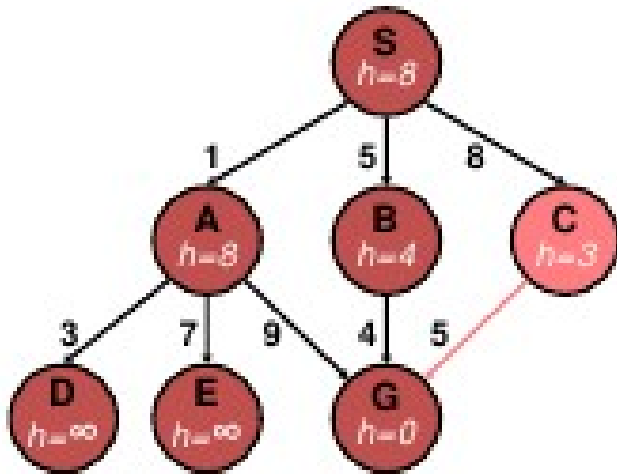
[S:8]

[C:3, B:4, A:8]

expanded

S

# Busca Gulosa: exemplo



frontier

expanded

[S:8]

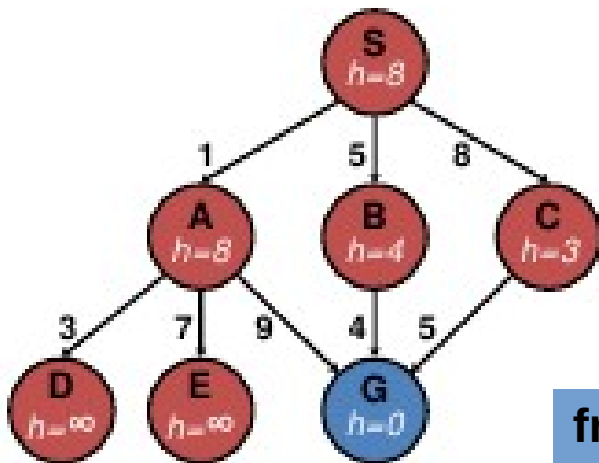
[C:3, B:4, A:8]

S

[G:0, B:4, A:8]

S, C

# Busca Gulosa: exemplo



frontier

[S:8]

[C:3, B:4, A:8]

[G:0, B:4, A:8]

[B:4, A:8]

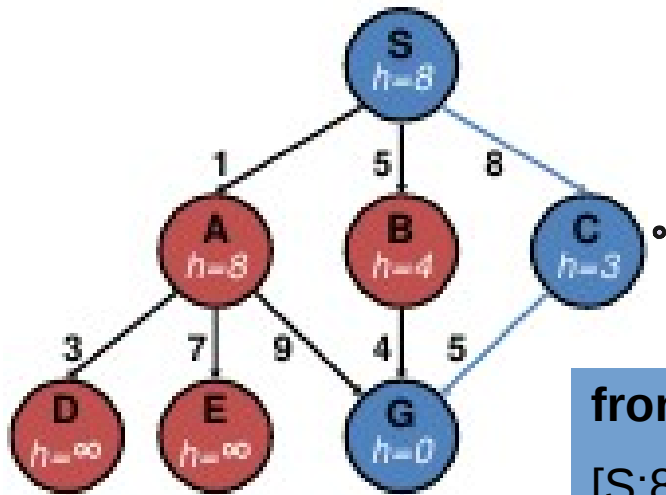
expanded

S

S, C

S, C, G

# Busca Gulosa: exemplo



Custo real da  
solução: 13

frontier

[S:8]

[C:3, B:4, A:8]

[G:0, B:4, A:8]

[B:4, A:8]

expanded

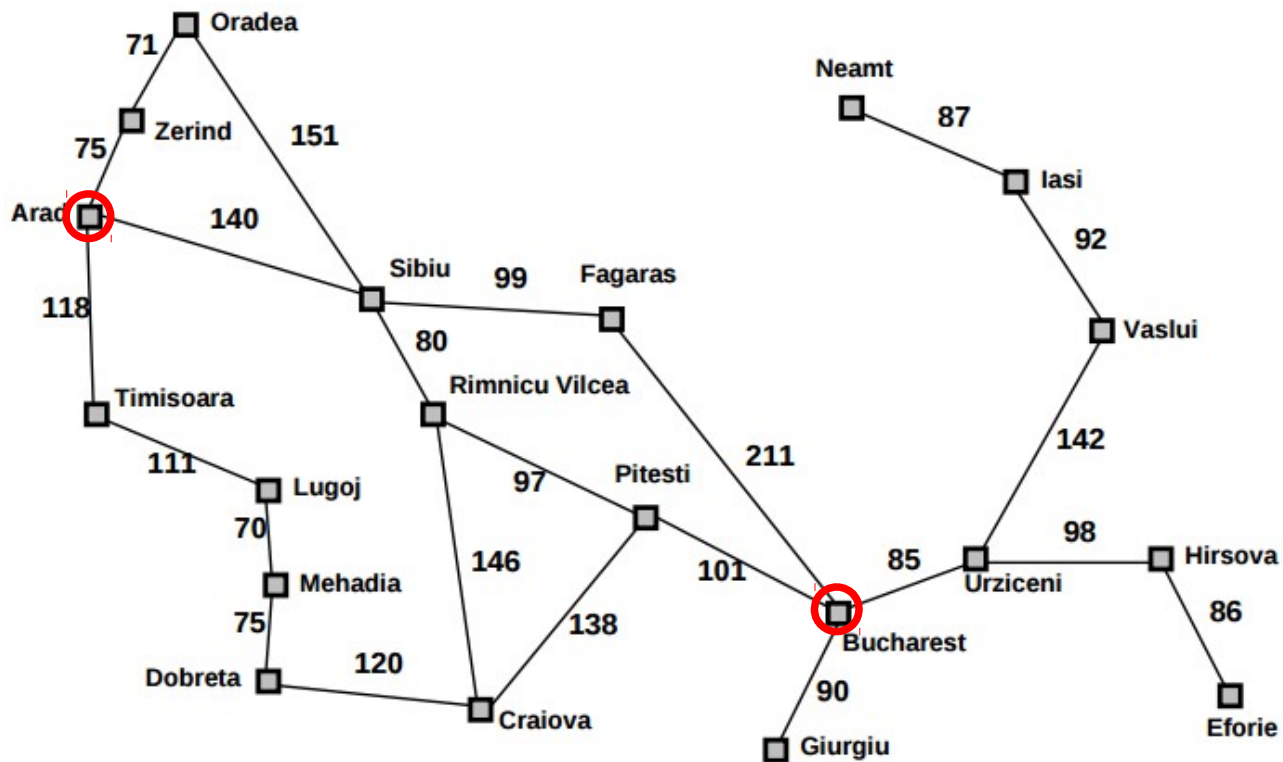
S

S, C

S, C, G

# Exemplo - Busca gulosa, de Arad a Bucharest

•  $h(n) = ???$





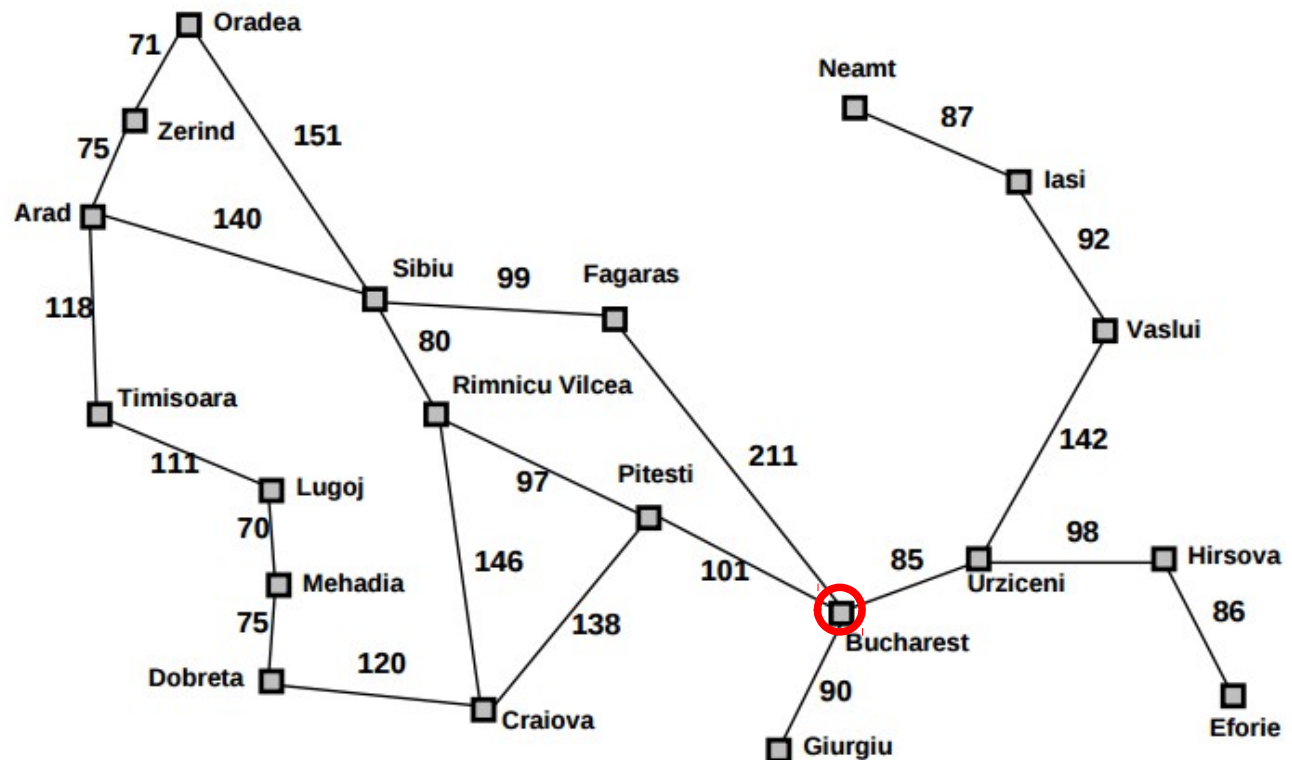
# Exemplo - Busca gulosa, de Arad a Bucharest

•  $h(n)$  = linha reta da cidade  $n$  até a meta

•  $D = ((X2 - X1)^2 + (Y2 - Y1)^2)^{1/2}$

Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

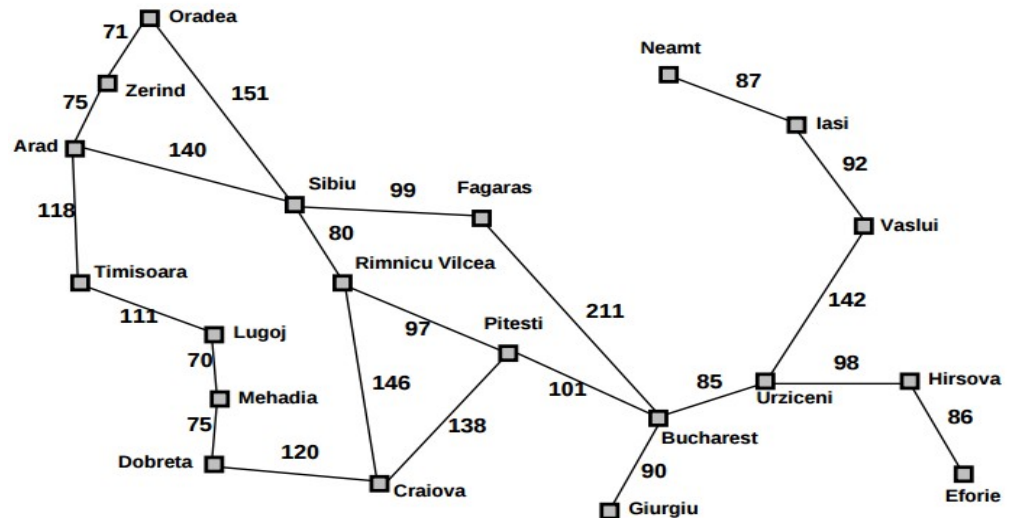


# 1 Exemplo de busca gulosa pela melhor escolha



Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

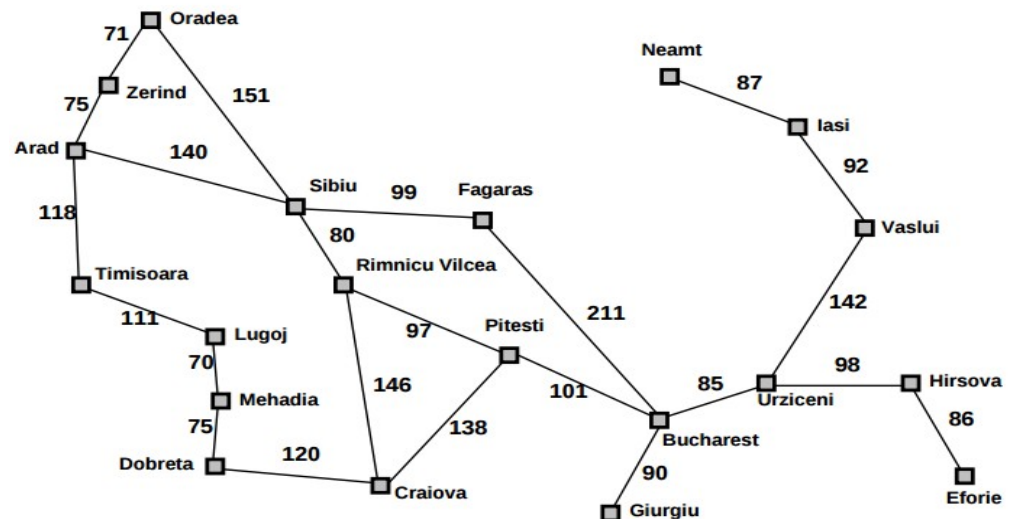


# Exemplo de busca gulosa pela melhor escolha



Straight-line distance  
to Bucharest

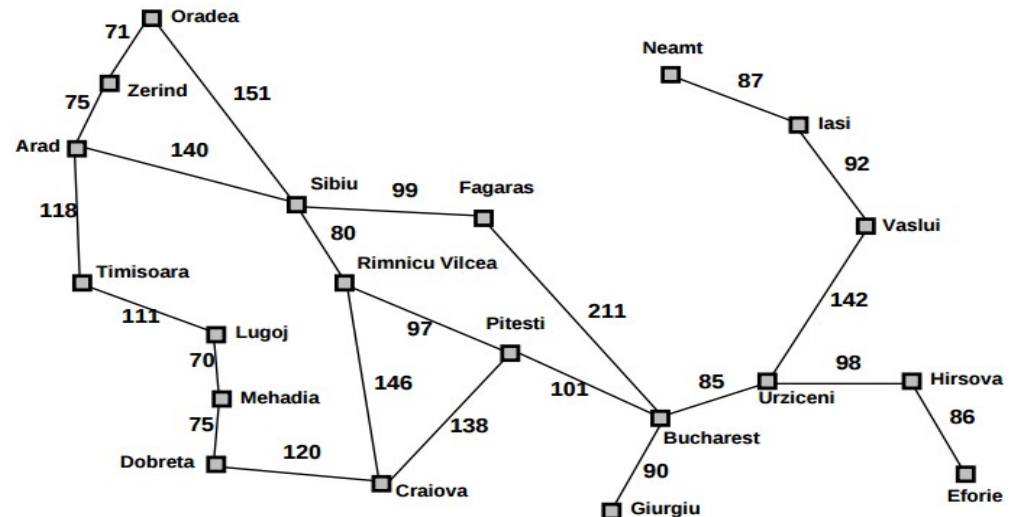
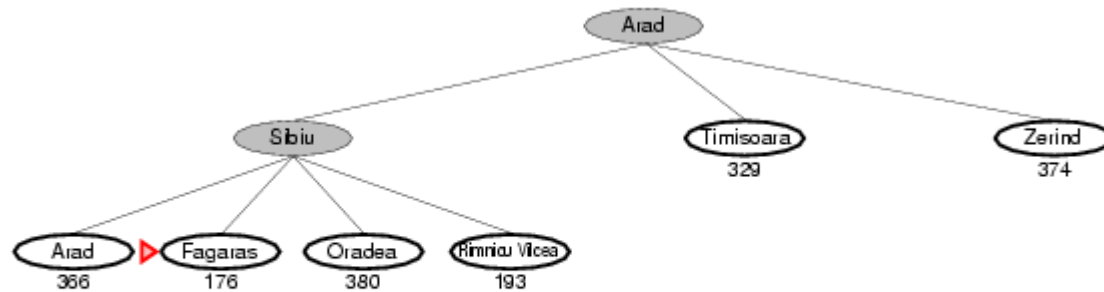
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



# Exemplo de busca gulosa pela melhor escolha

Straight-line distance  
to Bucharest

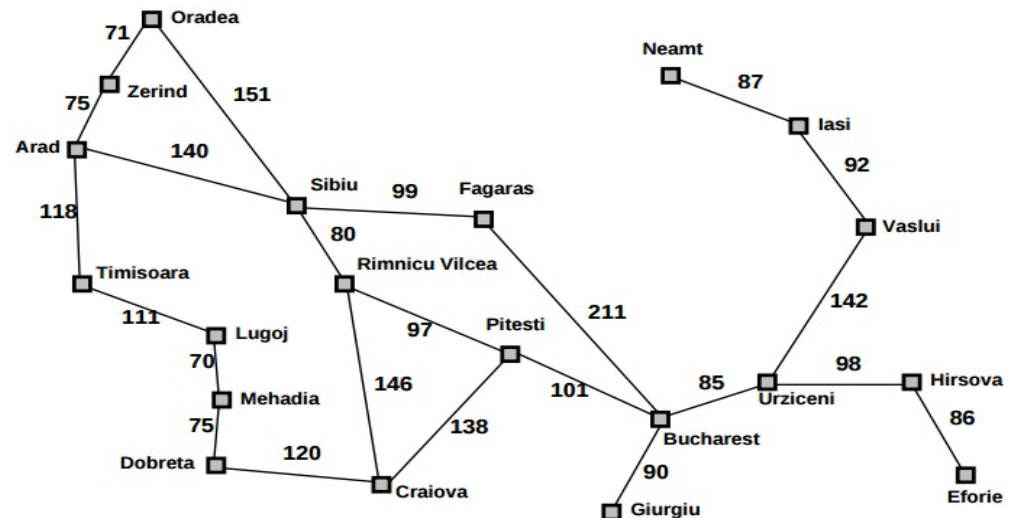
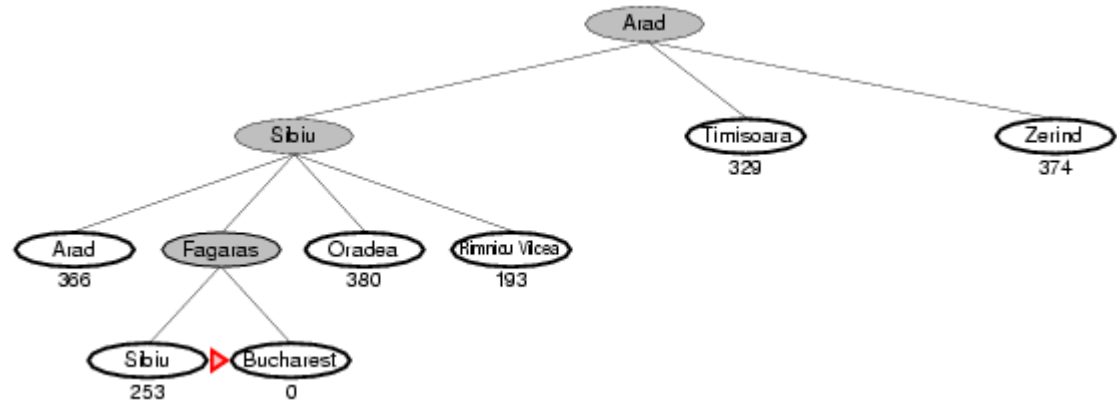
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



# Exemplo de busca gulosa pela melhor escolha

Straight-line distance  
to Bucharest

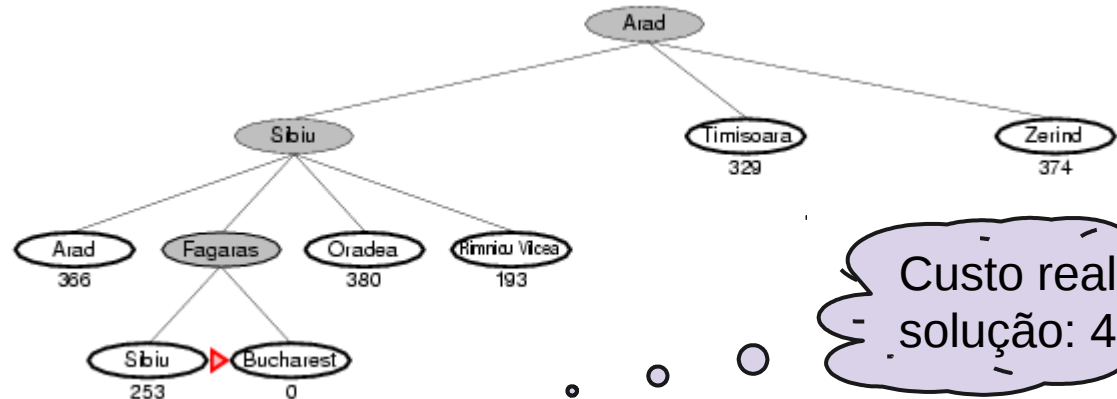
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



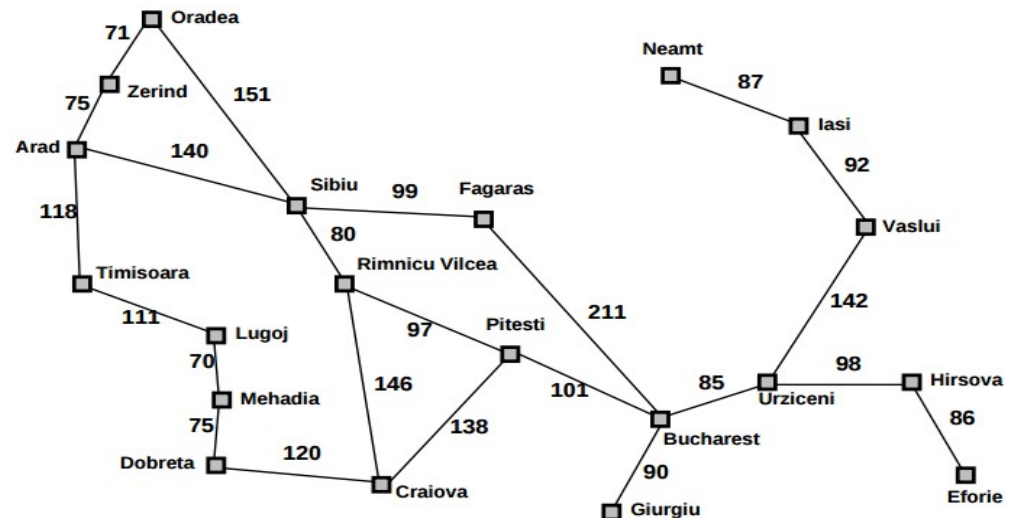
# Exemplo de busca gulosa pela melhor escolha

Straight-line distance  
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



Custo real da  
solução: 450



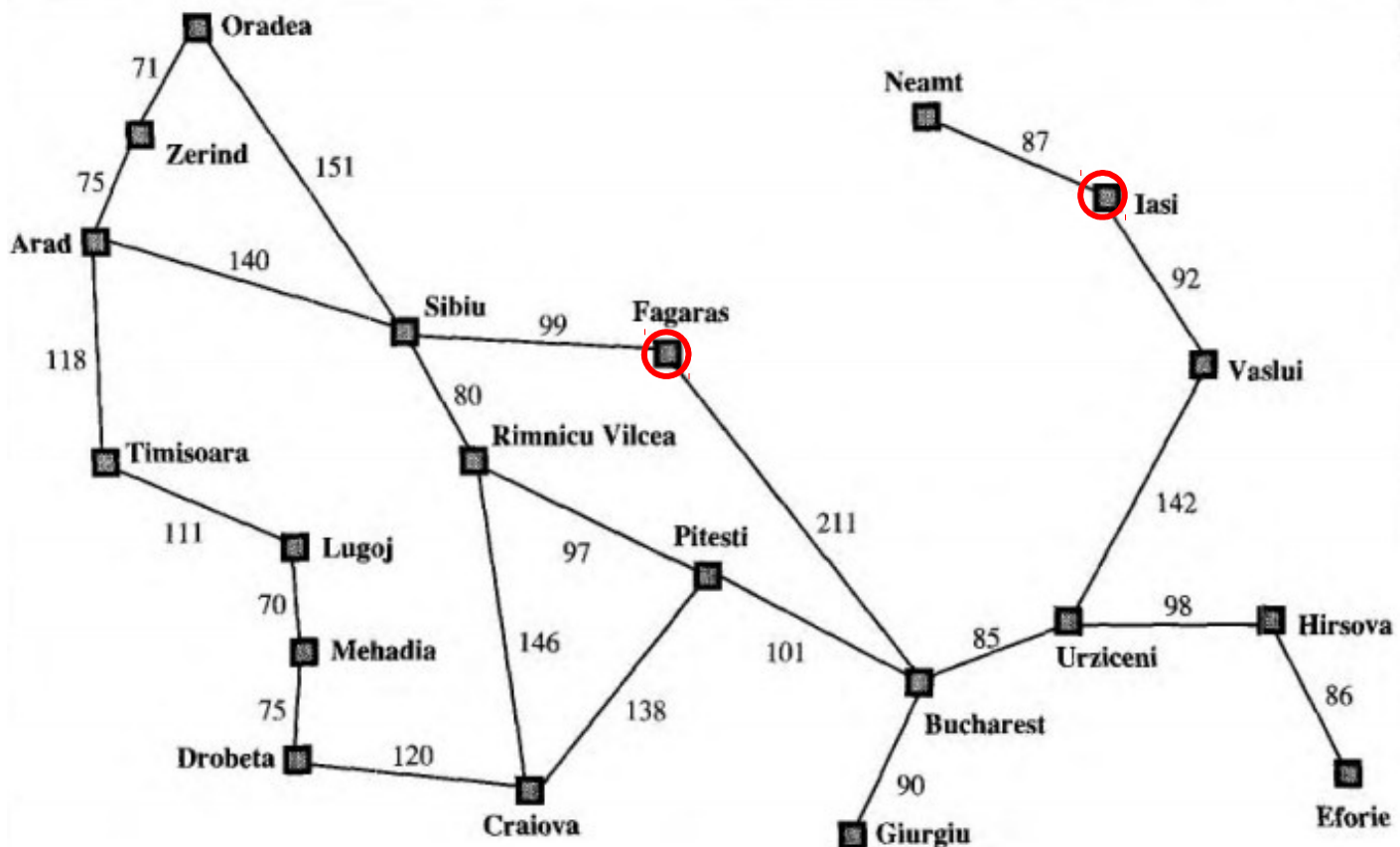


# Busca Gulosa

- **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 (418).
- **Não é completa**

# Busca Gulosa

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

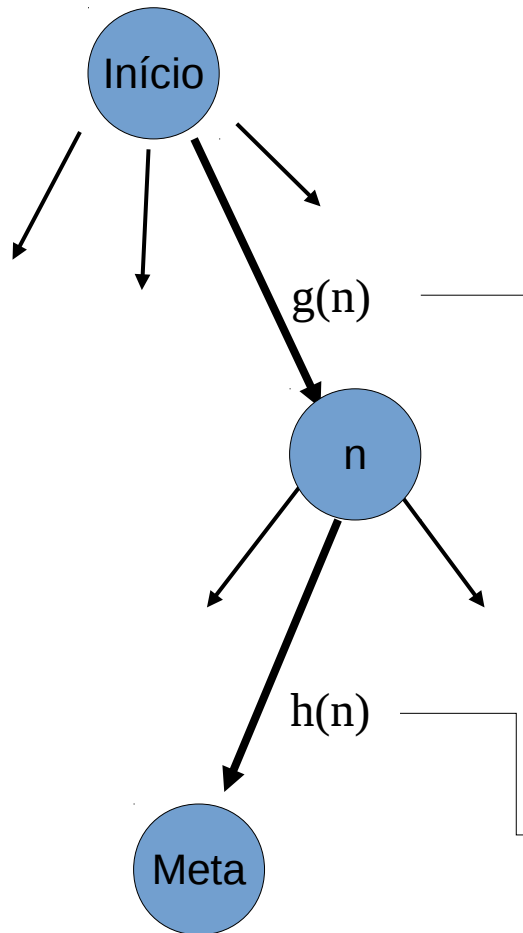




# Busca A\*

- Usa função de avaliação  $f(n) = h(n) + g(n)$ 
  - $h(n)$ : heurística, custo **estimado** do nó  $n$  até a meta
  - $g(n)$ : custo do estado inicial até o nó  $n$
  - $f(n)$ : custo estimado da solução que passa por  $n$

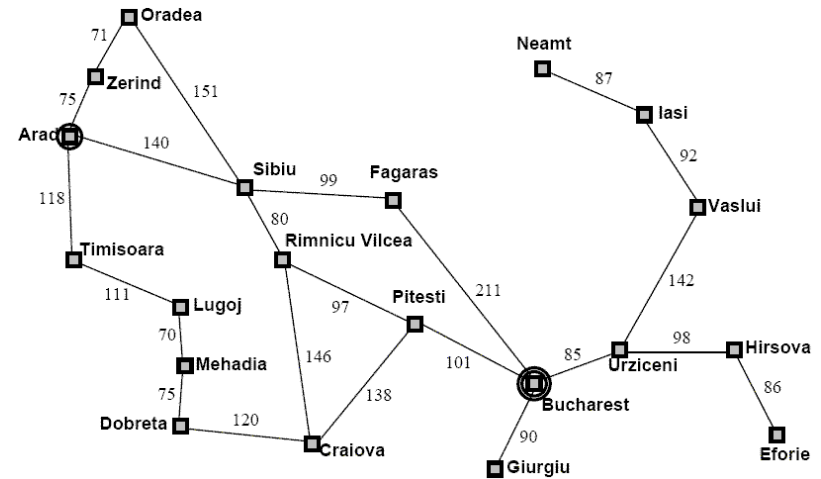
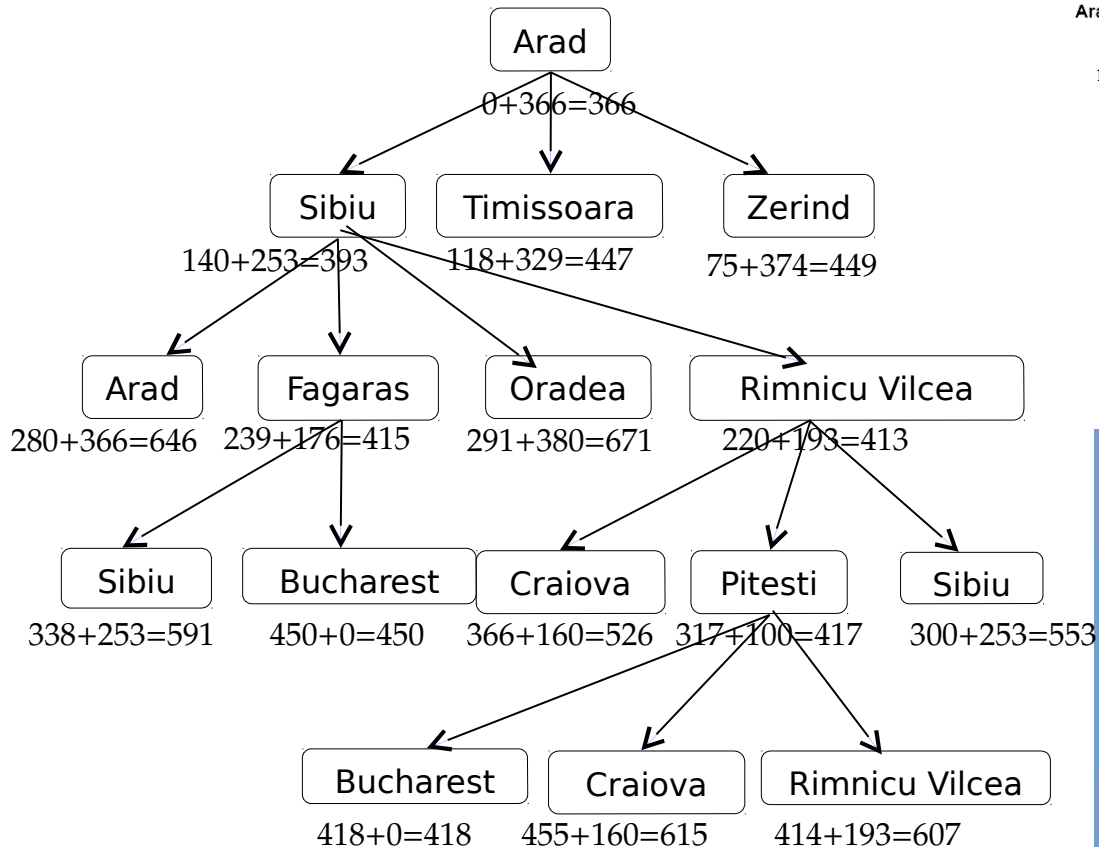
# Busca Heurística: função de avaliação



Podemos usar o custo do caminho  $g(n)$  para decidir que nó expandir como na Busca de Custo Uniforme. Mas esta medida não dirige a busca para o nó objetivo, ela cuida do **passado**.

A função heurística  $h(n)$  calcula o custo estimado para se alcançar o nó objetivo. Ela cuida do **futuro**.

# 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

# Como criar heurísticas ?

- Quebra cabeça de 8

7	2	4
5		6
8	3	1

Início

1	2	3
4	5	6
7	8	

Objetivo

# Como criar heurísticas ?

- Quebra cabeça de 8

$h_1(n)$ : número de peças fora da posição

7	2	4
5		6
8	3	1

Início

1	2	3
4	5	6
7	8	

Objetivo

# Como criar heurísticas ?

## <sup>1</sup> Quebra cabeça de 8

- <sup>1</sup>  $h2(n)$ : para cada quadrado, calcular a distância em quadrados até a sua posição: **distância de Manhattan**

7	2	4
5		6
8	3	1

Início

1	2	3
4	5	6
7	8	

Objetivo

# Como criar heurísticas ?

- Quebra cabeça de 8

- $h1(n)$ : número de peças fora da posição

- $h2(n)$ : distância de Manhattan

7	2	4
5		6
8	3	1

Início

1	2	3
4	5	6
7	8	

Objetivo

- $h1(n)$  ?

- $h2(n)$  ?

# Como criar heurísticas ?

- Quebra cabeça de 8

- $h1(n)$ : número de peças fora da posição

- $h2(n)$ : distância de Manhattan

7	2	4
5		6
8	3	1

1	2	3
4	5	6
7	8	

- $h1(S)$ : 6

- $h2(S)$ :  $4+0+3+3+1+0+2+1 = 14$



# Leitura Complementar

- Russell, S. and Norvig, P. **Artificial Intelligence: a Modern Approach**, 3rd Edition, Prentice-Hall, 2009.
- **Capítulo 4: Informed Search and Exploration**

