

Estratégias de Busca

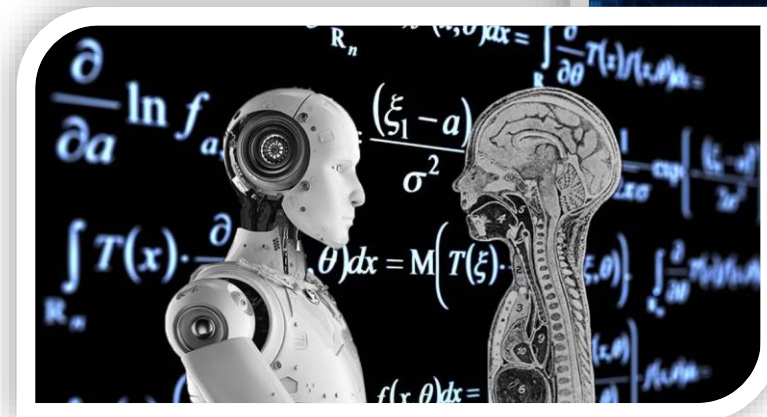
Disciplina: Inteligência Computacional (C210A/B)

Curso: Engenharia de Computação e Software

Prof^a. Victoria Dala Pegorara Souto

ESTRATÉGIAS DE BUSCA INFORMADA (HEURÍSTICA)

- Busca Gulosa
- Busca A*



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- ◎ Utiliza conhecimento de um problema específico além da definição do problema em si.
- ◎ Pode encontrar soluções de forma mais eficiente do que uma estratégia de busca sem informação (cega).

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Uma busca heurística é uma busca que utiliza uma função $h(n)$ onde, para cada nó n do espaço de busca, fornece uma **avaliação do custo** para atingir o estado final. Portanto, $h(n)$ é chamada função heurística.

$h(n)$ = custo estimado do caminho de menor custo do estado do nó n para um estado objetivo

- Note que $h(n)$ recebe um nó como entrada e **depende apenas do estado naquele nó**.
- Funções heurísticas são a forma mais comum como o conhecimento adicional do problema é transmitido ao algoritmo de busca.
- Funções Heurísticas são **específicas** para cada problema.

Roteamento: $h(n)$ = rota mais barata de Corumbá a Campo Grande;

Árvore: $h(n)$ = distância direta entre n e o nó final.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- ◎ Os problemas de IA empregam heurísticas, basicamente, em duas situações:
 - Um problema pode **não ter uma solução exata** por causa das ambiguidades inerentes a sua formulação ou pela (in)disponibilidade dos dados.
 - **Exemplos:** Diagnóstico médico, Sistemas de visão.
 - Um problema pode ter uma solução exata, mas o **custo computacional** para encontrá-la pode ser **proibitivo**.
 - **Exemplos:** Jogo de xadrez, Caixeiro viajante.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Exemplo:

2	8	3
1	6	4
7		5

Estado Inicial

1	2	3
8		4
7	6	5

Estado Objetivo

**Como definir uma Função
Heurística $h(n)$?**

**Existe apenas uma
solução?**

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Exemplo:

2	8	3
1	6	4
7		5

Estado Inicial

Solução 1:

$h(n)$ = Número de elementos em posições erradas (com relação ao estado objetivo).

$$h(n) = 4$$

1	2	3
8		4
7	6	5

Estado Objetivo

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Exemplo:

2	8	3
1	6	4
7		5

Estado Inicial

Solução 2:

$h(n)$ = Soma da distância de cada elemento (d_n) à posição final.

$$h(n) = \sum_{n=1}^N d_n$$

$N \rightarrow$ Número de elementos

1	2	3
8		4
7	6	5

Estado Objetivo

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Exemplo:

1 casa para a direita - $d_n = 1$

2	8	3
1	6	4
7		5

Estado Inicial

Solução 2:

$h(n)$ = Soma da distância de cada elemento (d_n) à posição final.

$$h(n) = \sum_{n=1}^N d_n$$

$N \rightarrow$ Número de elementos

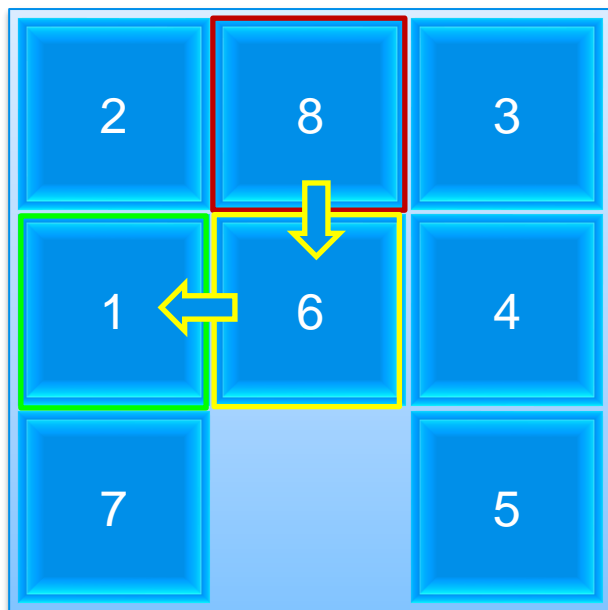
1	2	3
8		4
7	6	5

Estado Objetivo

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Exemplo:

2 casas - $d_n = 2$



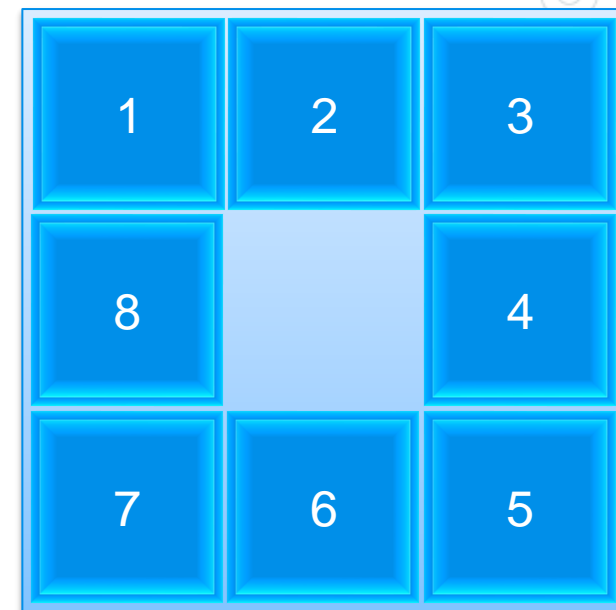
Estado Inicial

Solução 2:

$h(n)$ = Soma da distância de cada elemento (d_n) à posição final.

$$h(n) = \sum_{n=1}^N d_n$$

$N \rightarrow$ Número de elementos



Estado Objetivo

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Exemplo:

2	8	3
1	6	4
7		5

Estado Inicial

Solução 2:

$h(n)$ = Soma da distância de cada elemento (d_n) à posição final.

$$h(n) = \sum_{n=1}^8 d_n = 1 + 1 + 0 + 0 + 0 + 1 + 0 + 2 = 5$$

1	2	3
8		4
7	6	5

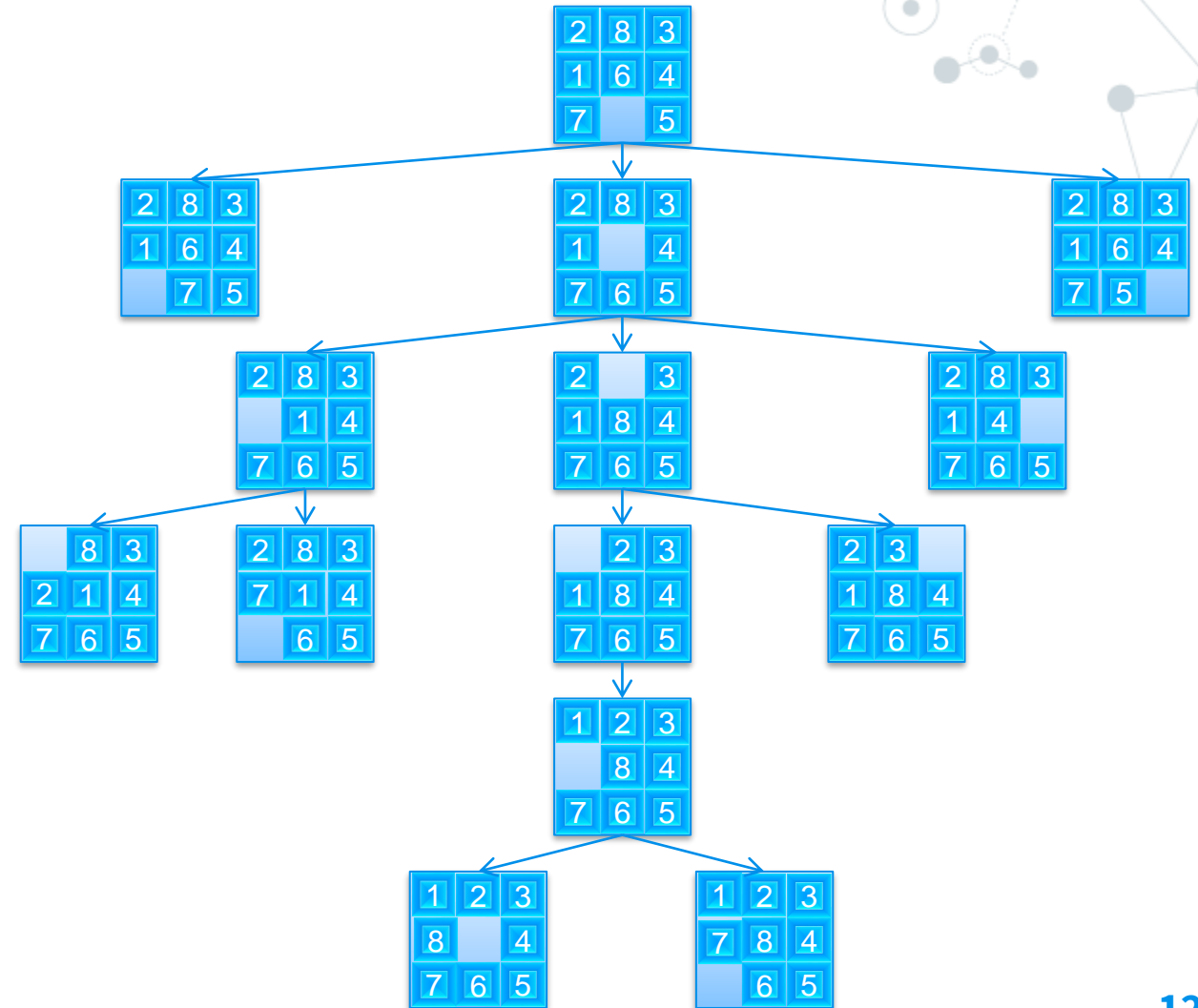
Estado Objetivo

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- ◎ **Exemplo:** Montando uma árvore para resolver o problema de busca.

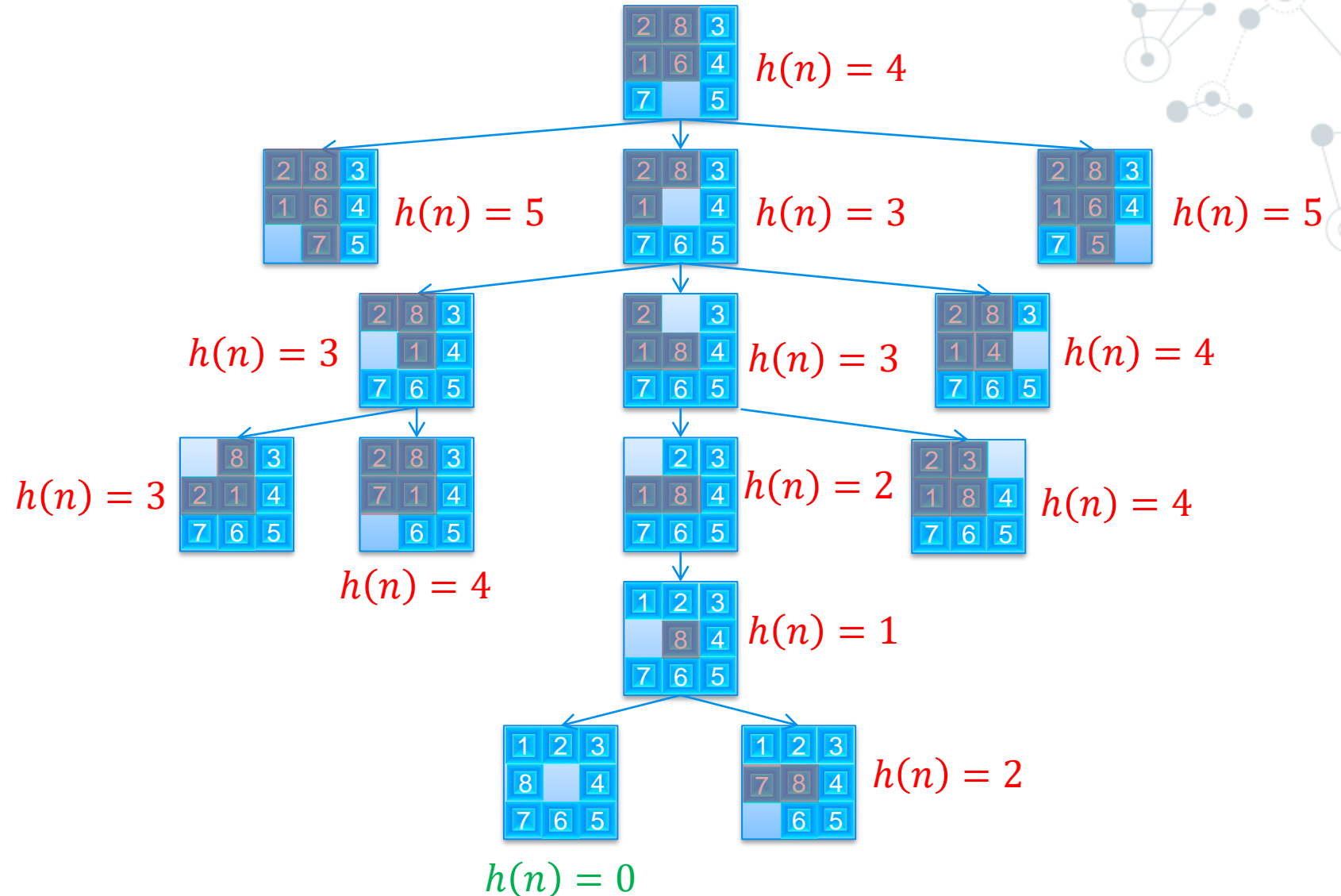
Considerando:

- $f(n)$ = Número de elementos em posições erradas.
- **Determine a solução de menor custo.**



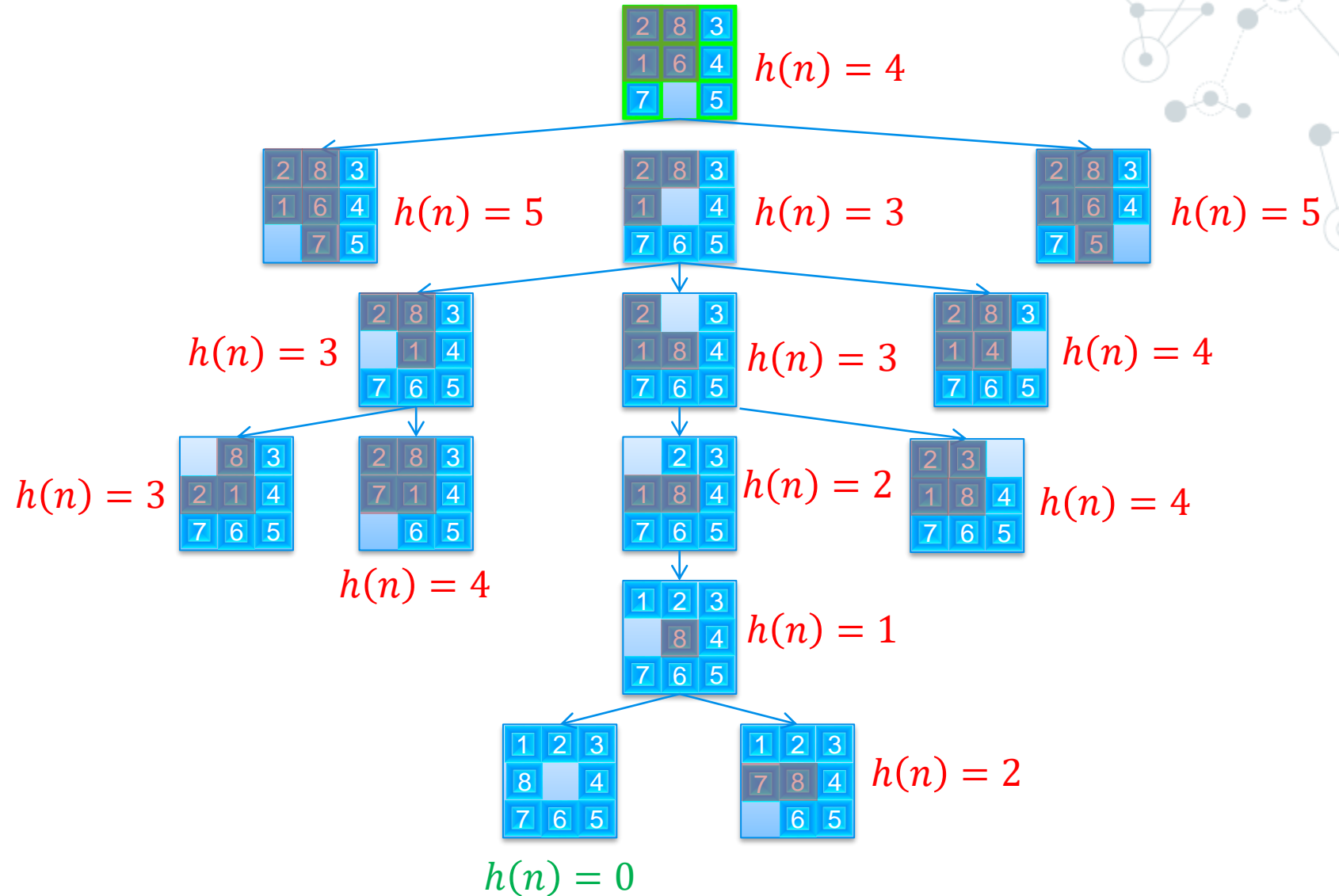
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



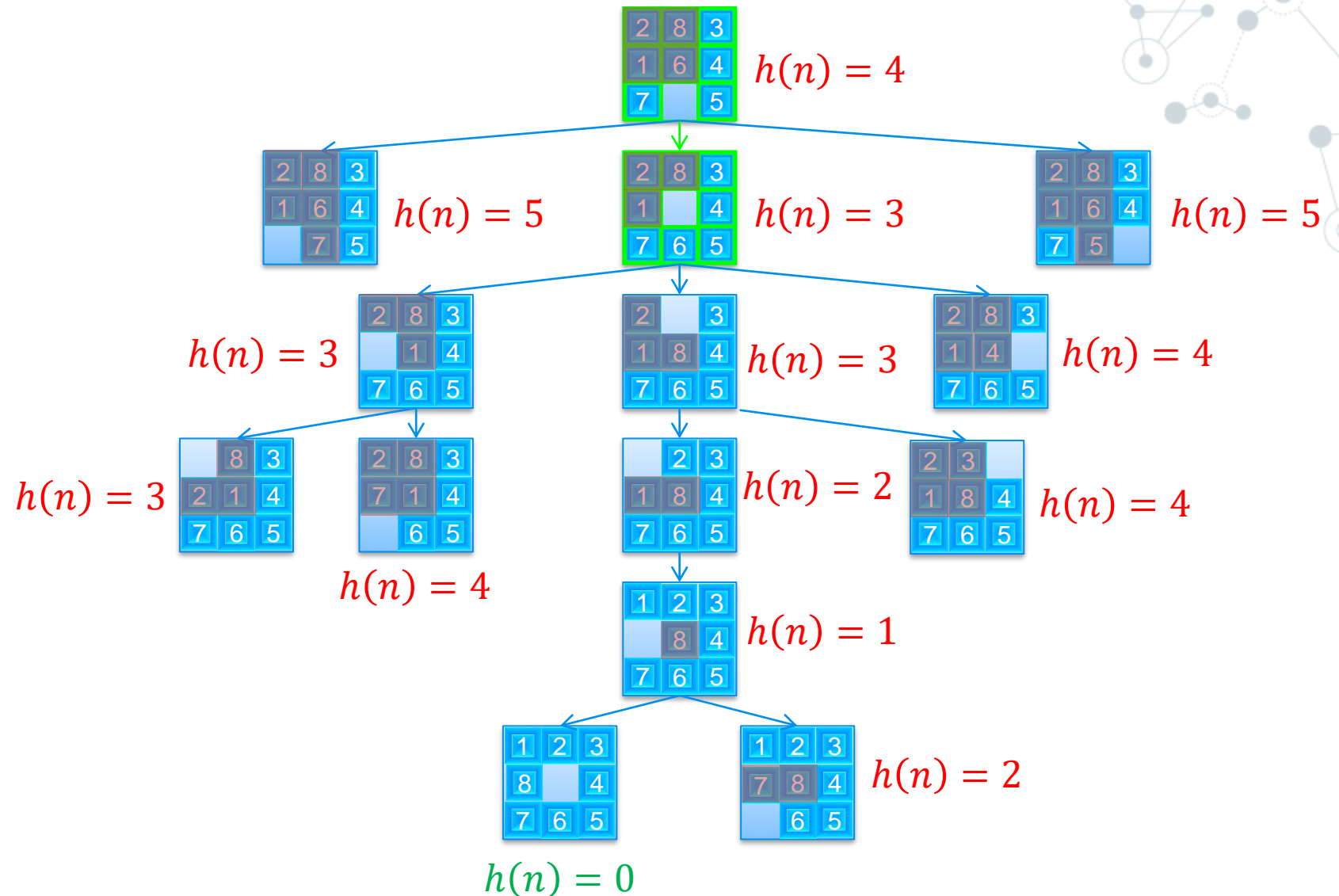
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



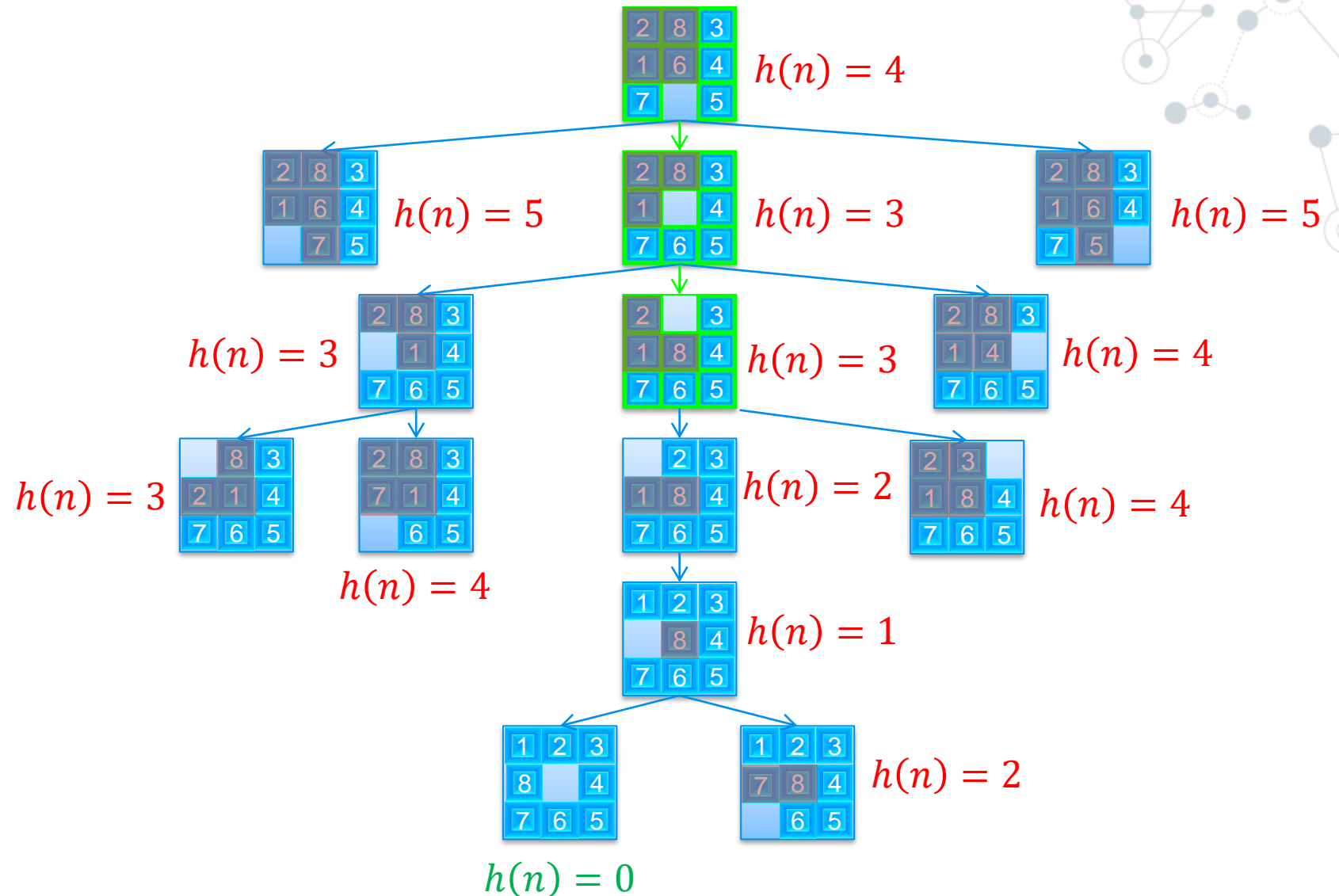
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



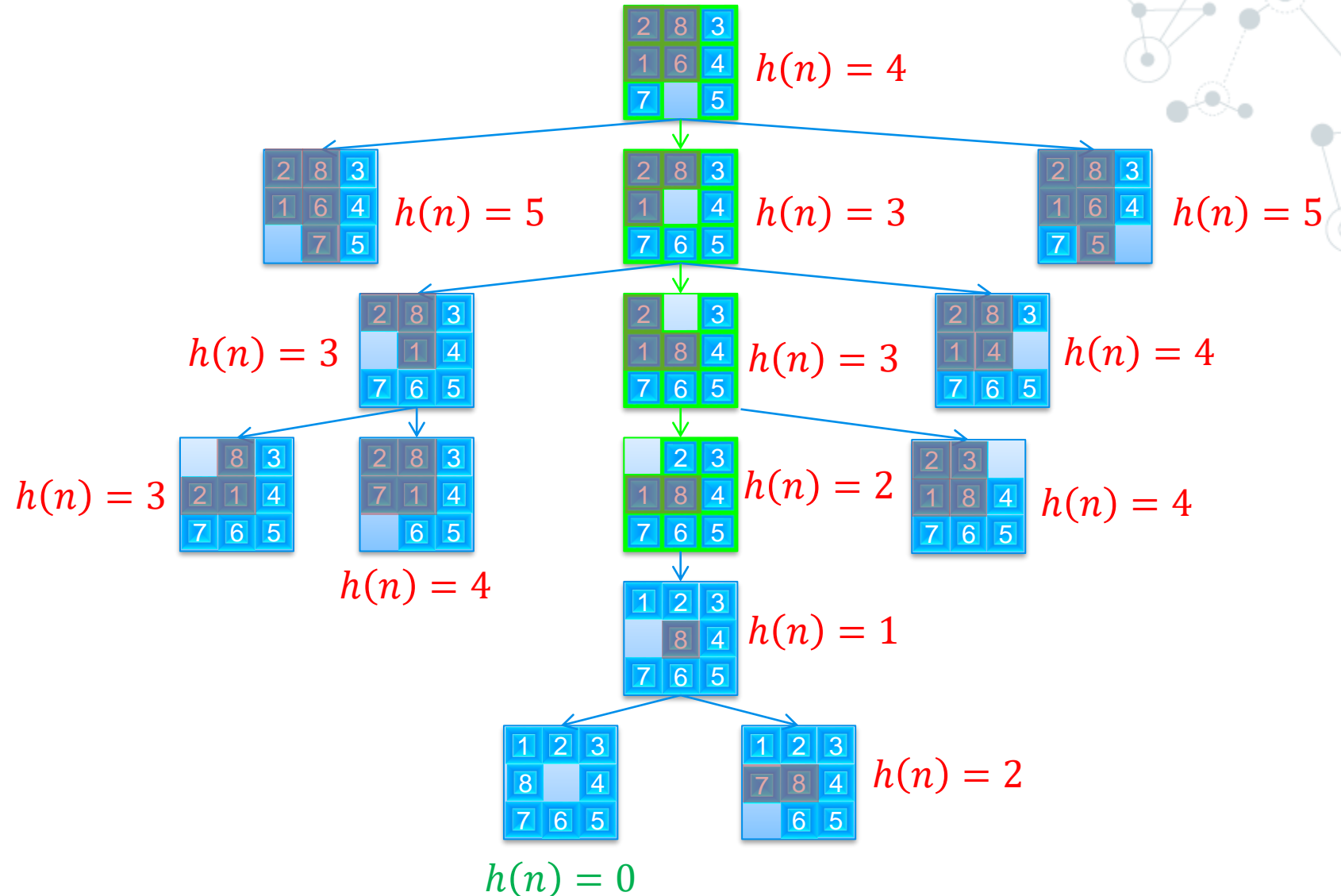
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



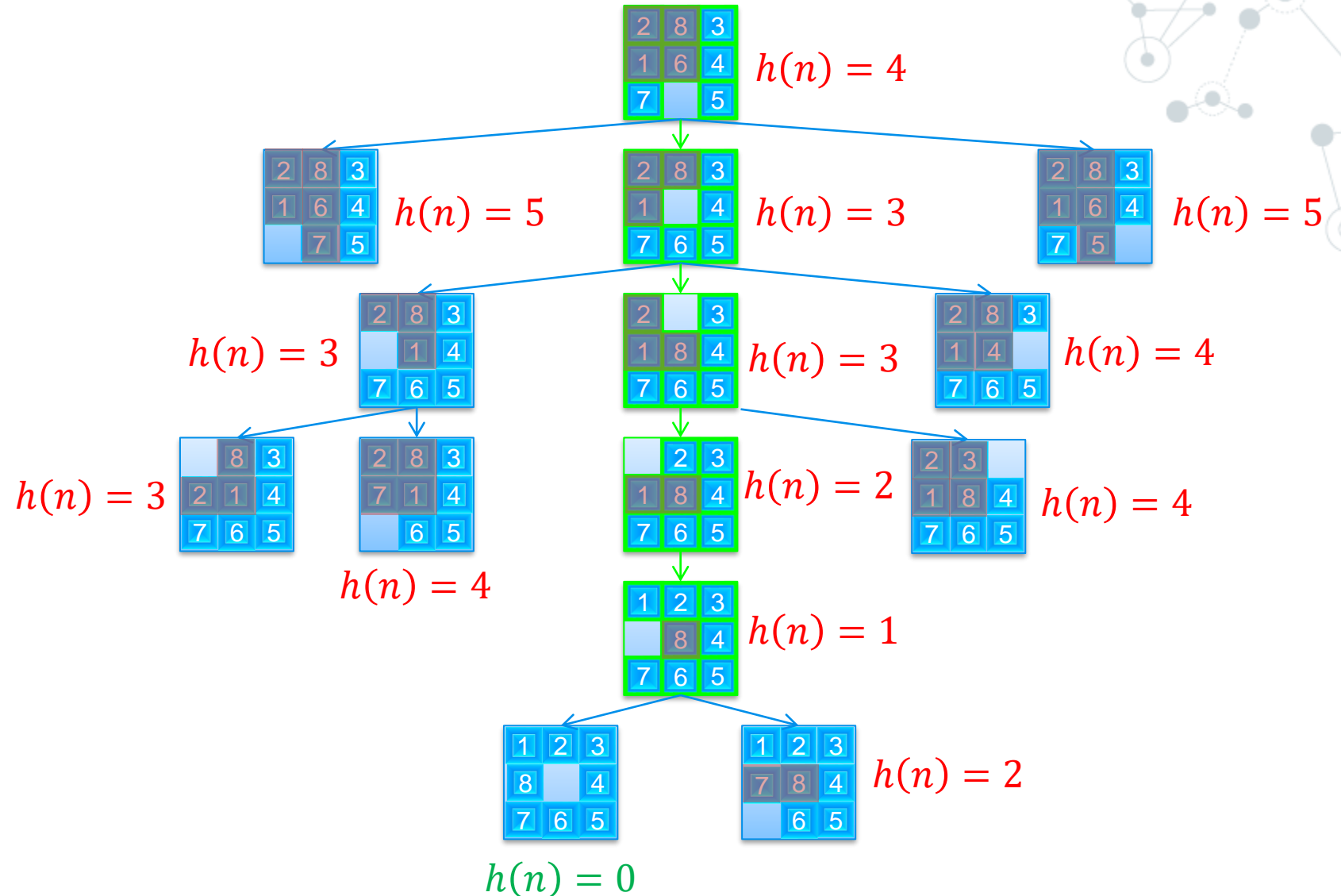
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



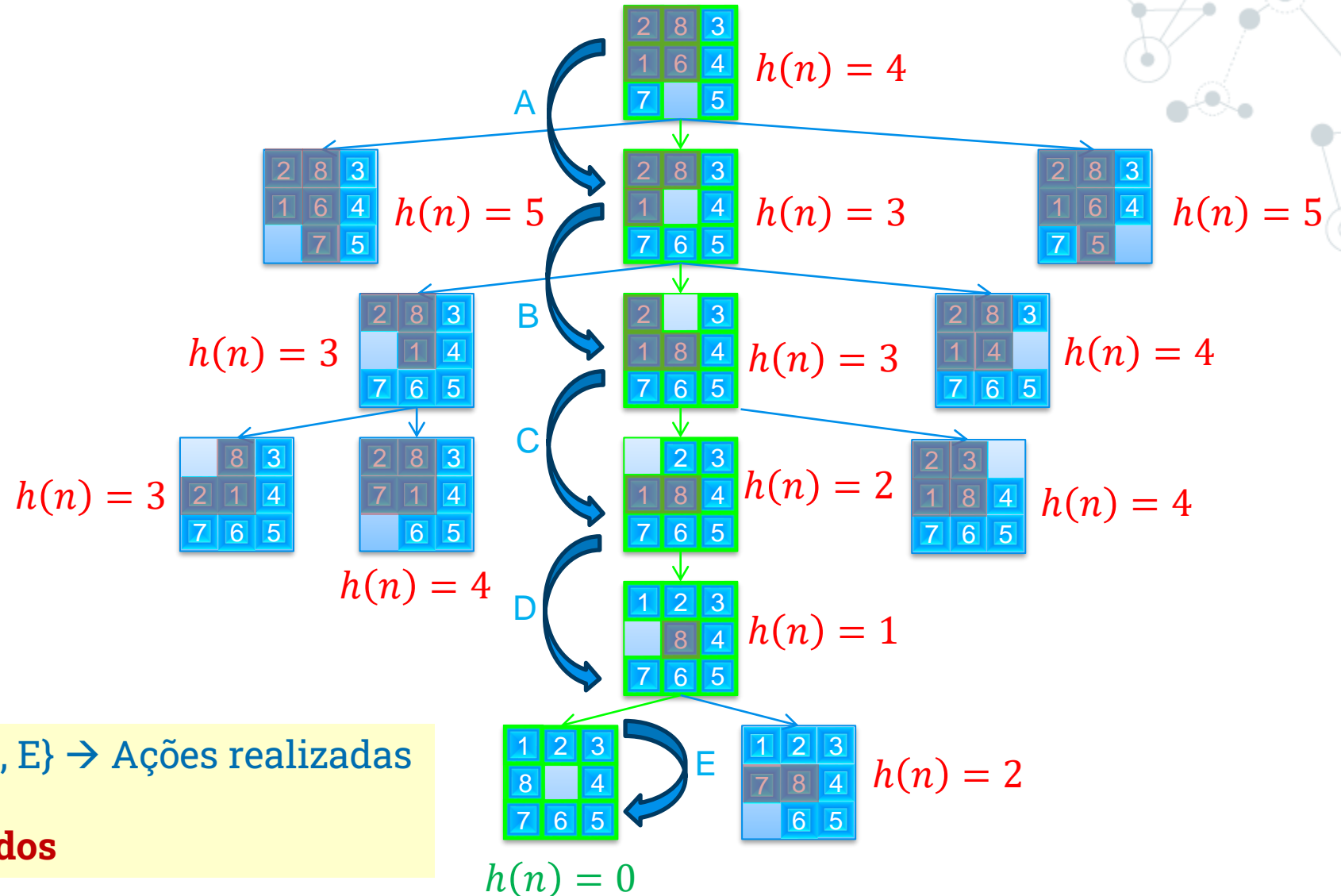
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ **Solução:**



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca Gulosa (Busca de Melhor Escolha)

- Expande o nó que está **mais próximo do objetivo**.
- Avalia os nós usando apenas a **função heurística**, ou seja, $f(n) = h(n)$.
- Pode conduzir a uma solução rapidamente.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca Gulosa (Busca de Melhor Escolha)

- **Exemplo:** Roteamento na Romênia.
 - **Objetivo:** Buscareste
 - **Função Heurística:** Distância em linha reta (DLR) definida por h_{DLR} .
 - É necessário conhecer as distâncias em linha reta para Bucareste.

Tabela 1 - h_{DLR} para todas as cidades

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

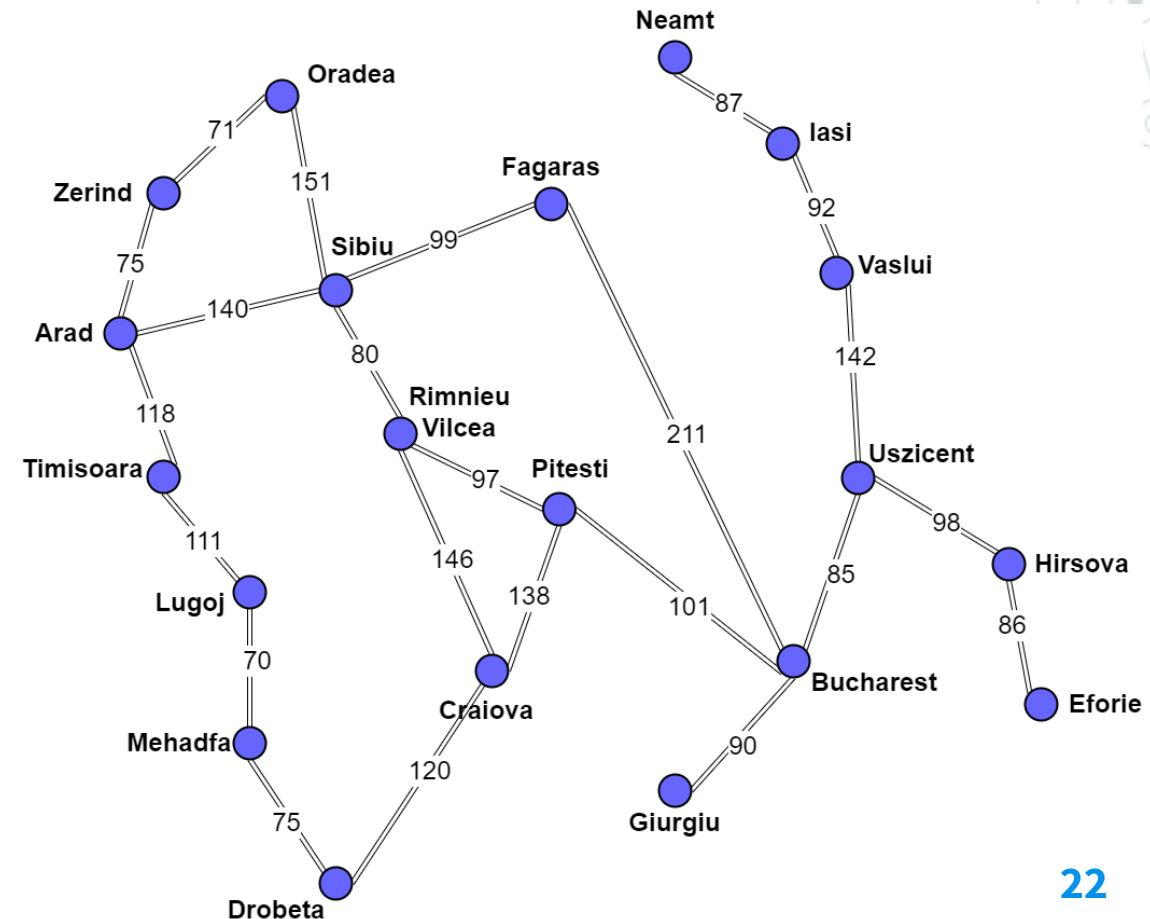
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca Gulosa (Busca de Melhor Escolha)

- **Exemplo:** Roteamento na Romênia.

Tabela 1 - h_{DLR} para todas as cidades

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



Busca Gulosa (Busca de Melhor Escolha)

- Exemplo:** Roteamento na Romênia.

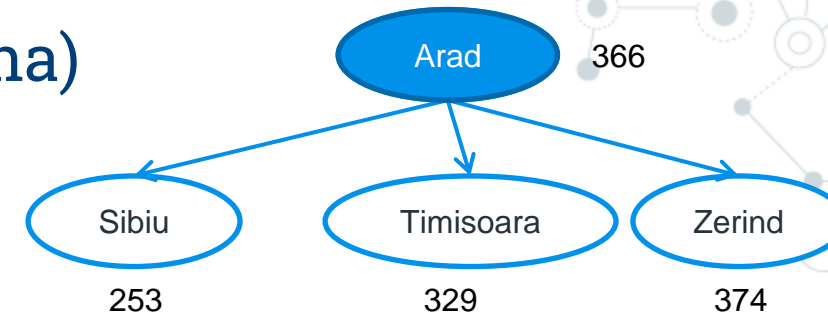
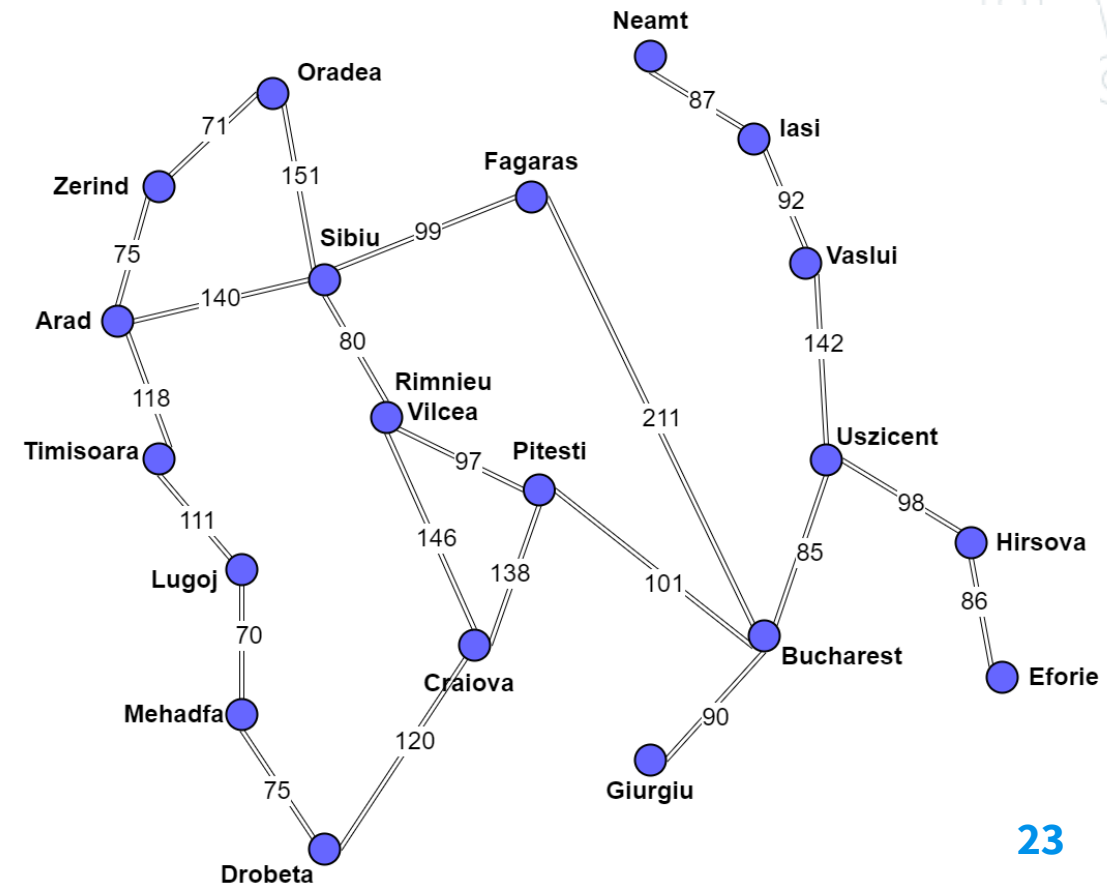


Tabela 1 - h_{DLR} para todas as cidades

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



Busca Gulosa (Busca de Melhor Escolha)

- Exemplo:** Roteamento na Romênia.

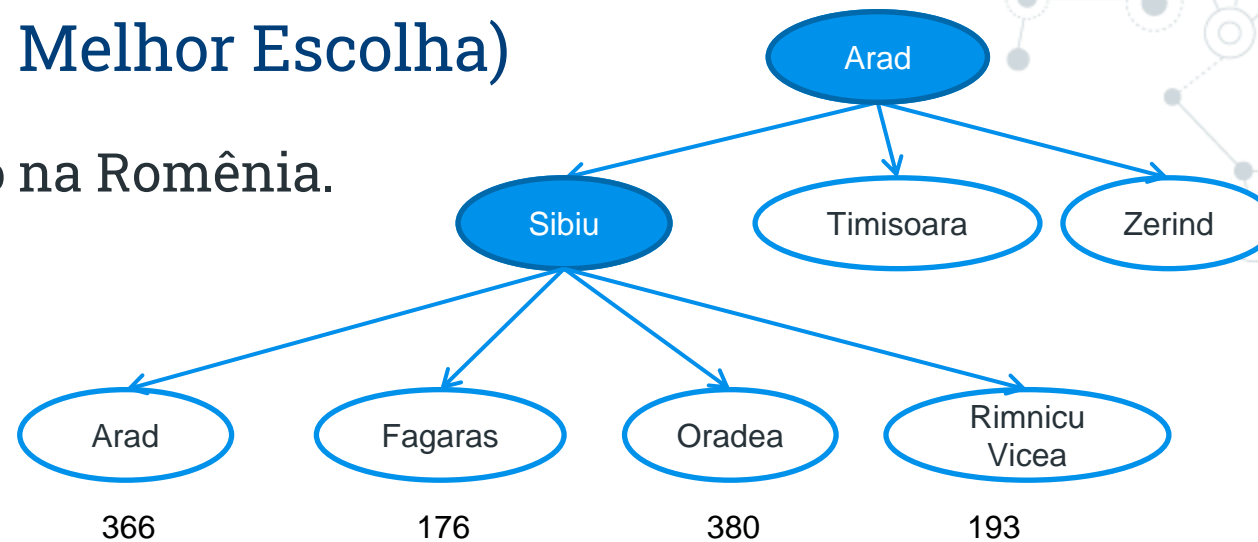
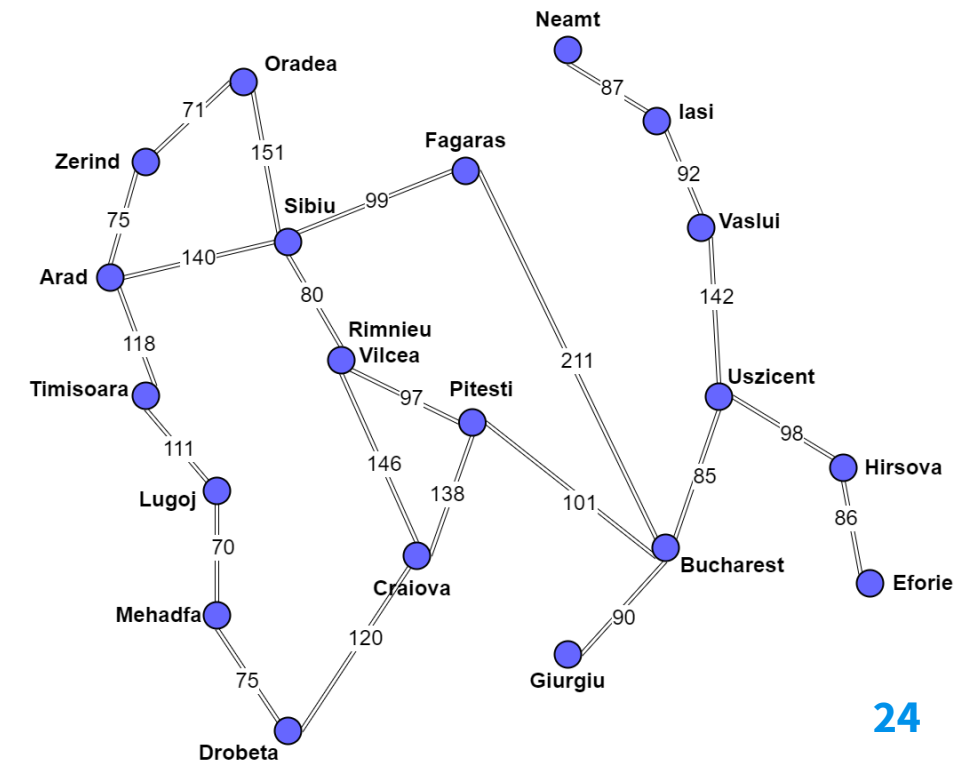


Tabela 1 - h_{DLR} para todas as cidades

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



Busca Gulosa (Busca de Melhor Escolha)

- Exemplo:** Roteamento na Romênia.

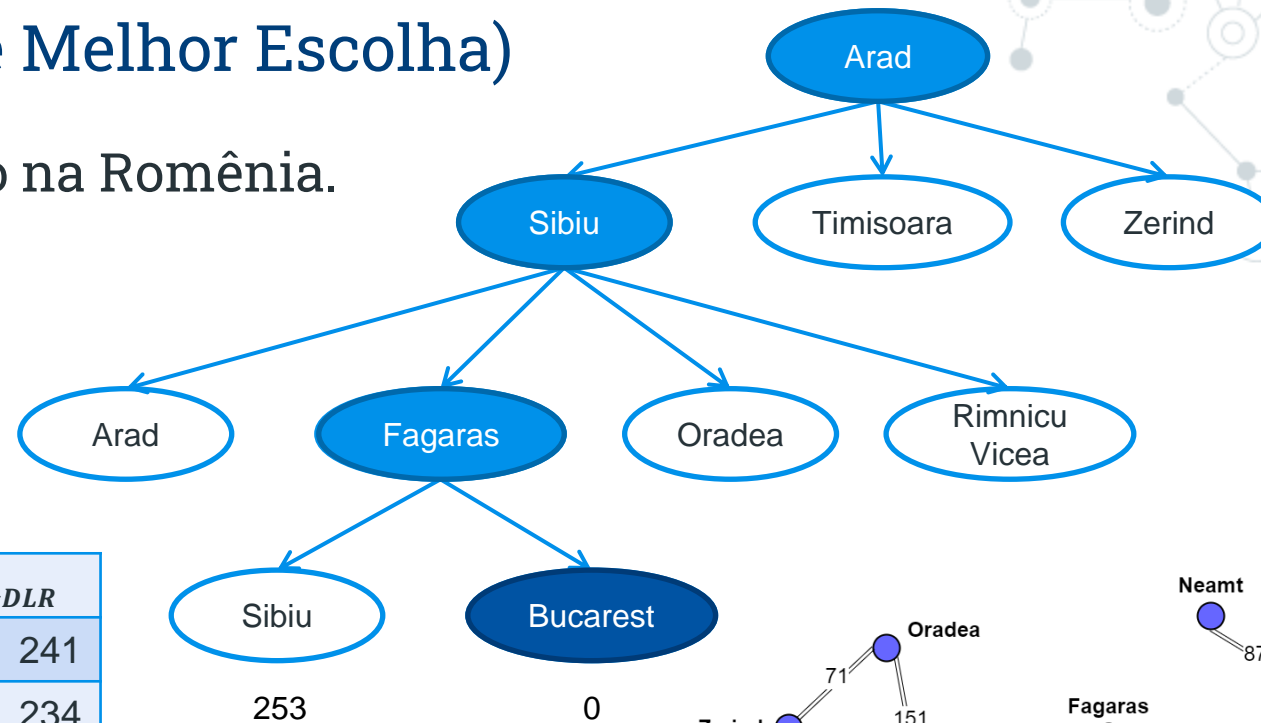
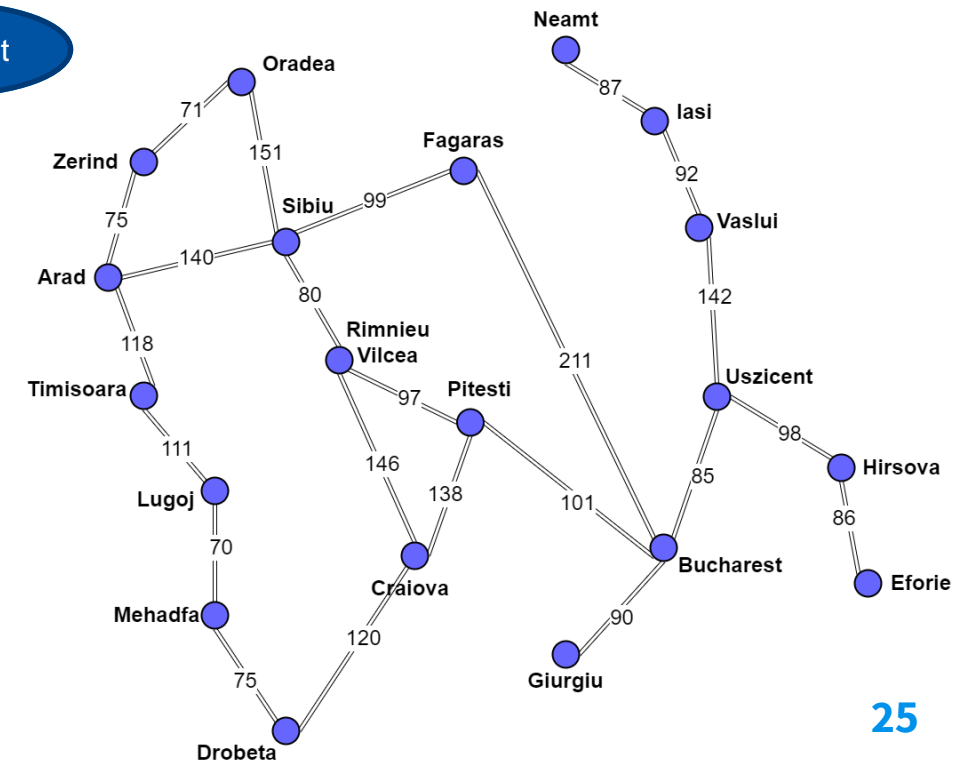


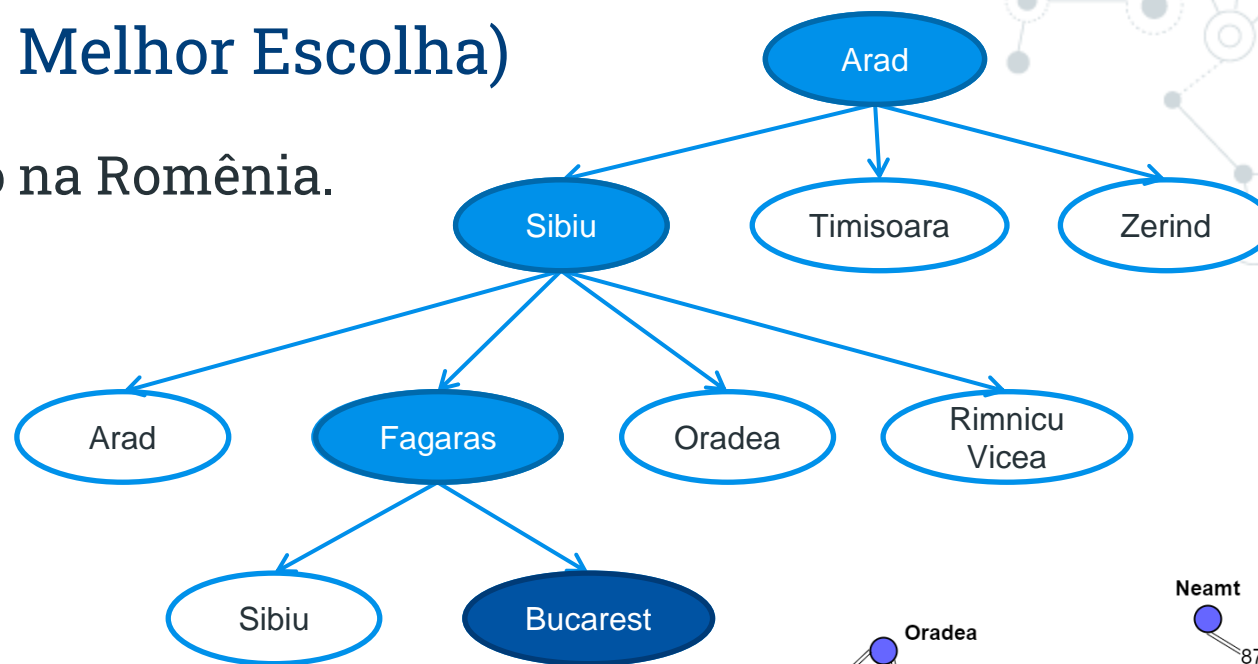
Tabela 1 - h_{DLR} para todas as cidades

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



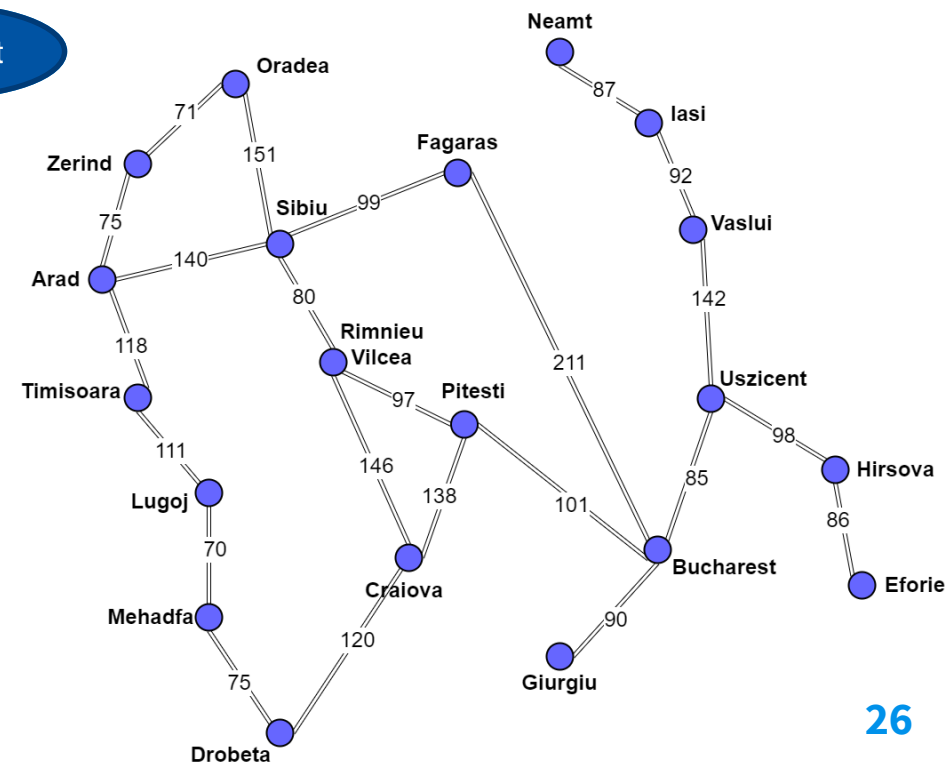
Busca Gulosa (Busca de Melhor Escolha)

- Exemplo:** Roteamento na Romênia.



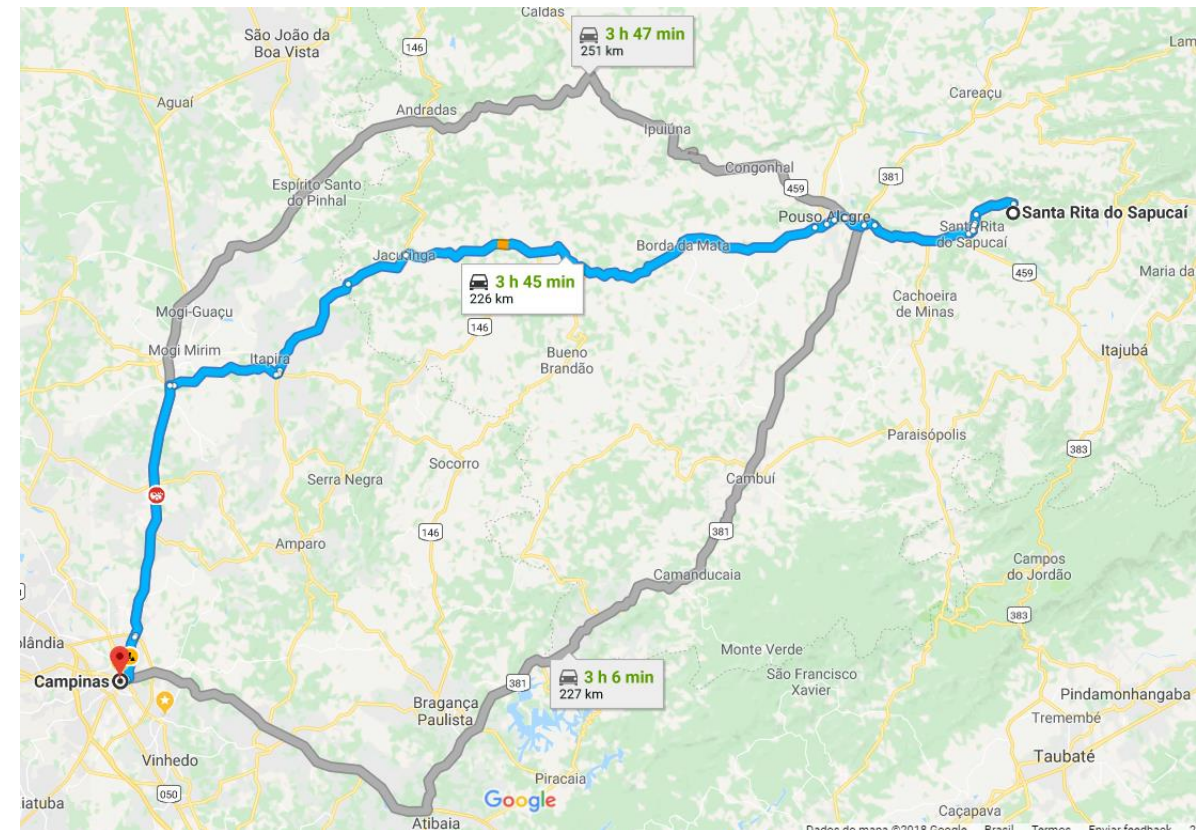
Observações:

- Para este exemplo, a busca gulosa utilizando h_{DLR} encontra uma solução, sem nunca expandir um nó que não estiver no caminho da solução.
- Seu custo de busca é **mínimo**.
- **NÃO é Ótimo**: O caminho via Sibiu e Fagaras para Bucareste é 32 km mais longo que o caminho através de Rimnicu Vilcea e Pitesti.
- **Algoritmo "ambicioso"** → A cada passo ele tenta chegar o mais próximo do destino que puder.
- A Busca Gulosa é **Incompleta** (igual a Busca em Profundidade)
Ex.: Iasi → Fagaras.



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- ◎ Busca Gulosa (Busca de Melhor Escolha)
 - **Exemplo:** Roteamento Santa Rita do Sapucaí -Campinas.

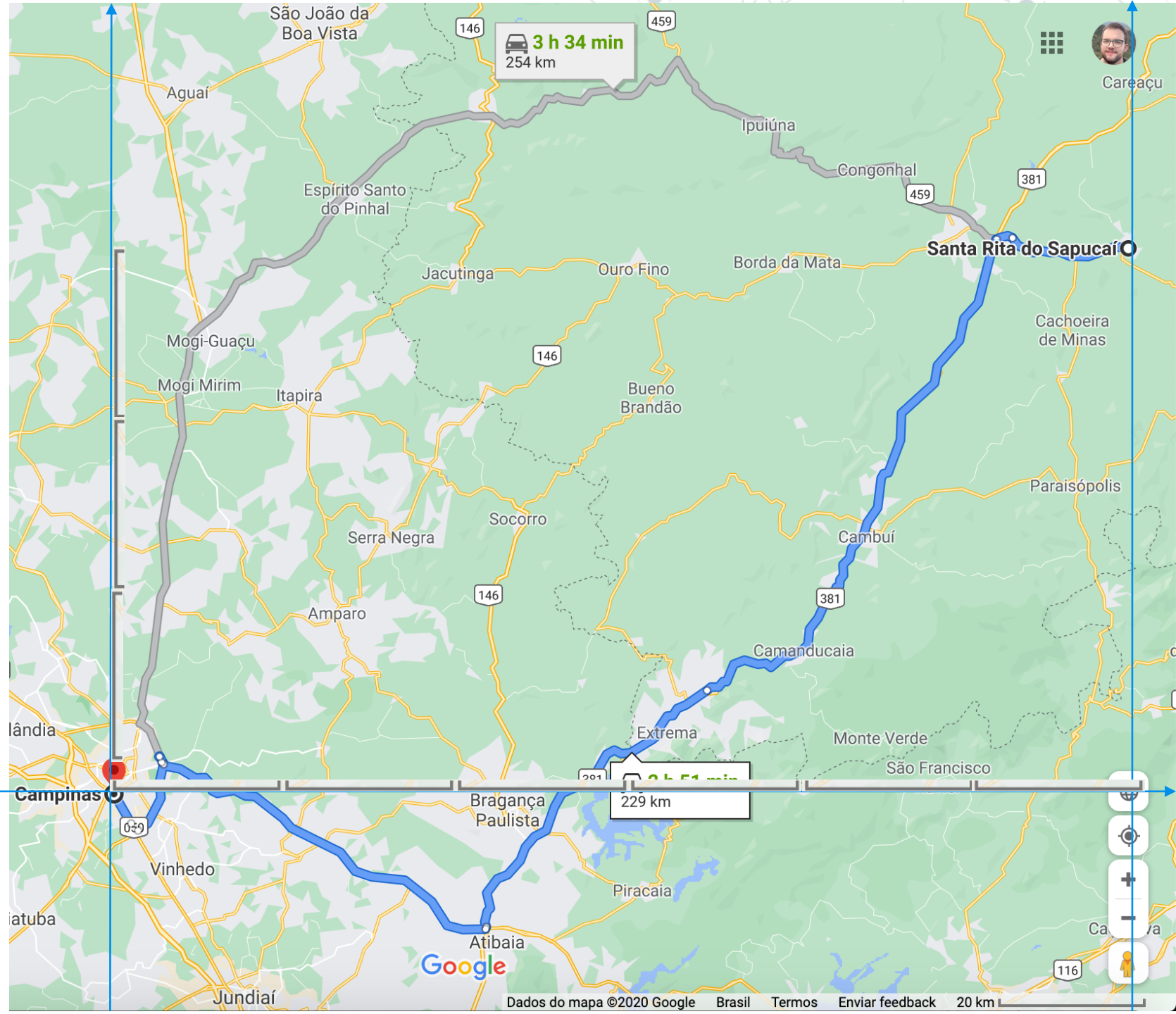


Busca Gulosa

- **Objetivo:** Campinas
- **Função Heurística:** Distância em linha reta (DLR) definida por h_{DLR} .
- É necessário conhecer as distâncias em linha reta para Campinas.

S. R. Sapucaí	165	Jacutinga	84
Pouso Alegre	137	Itapira	58
Cambuí	108	Congonhal	135
Camanducaia	97	Ipuiúna	139
Bragança Paulista	54	Andradas	106
Atibaia	57	Esp. Santo Pinhal	86
Campinas	0	Mogi-Guaçu	62
Borda da Mata	117	Mogi Mirim	54

Fonte: <https://www.adistanciaentre.com>



- Busca Gulosa

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas

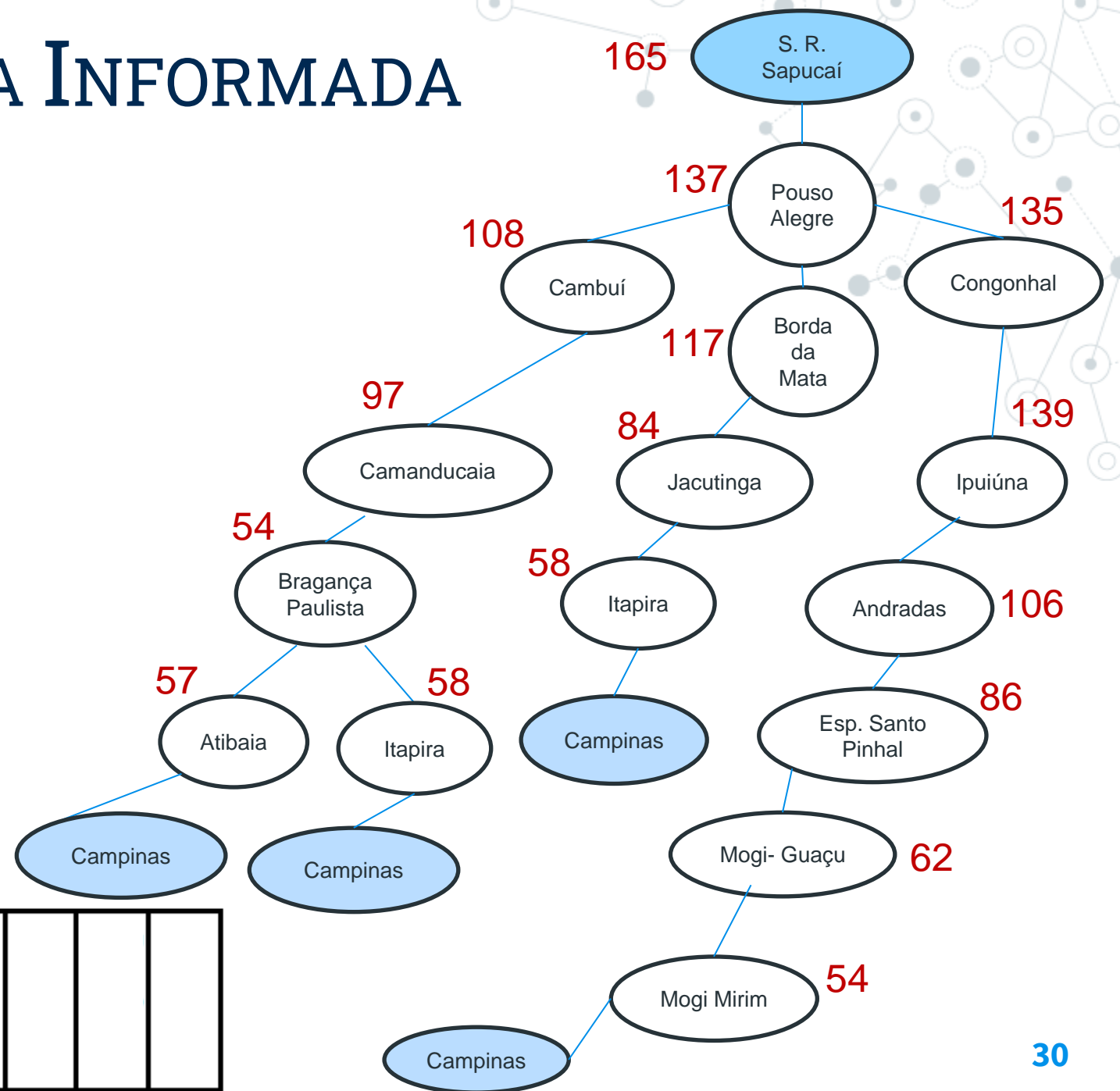


ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

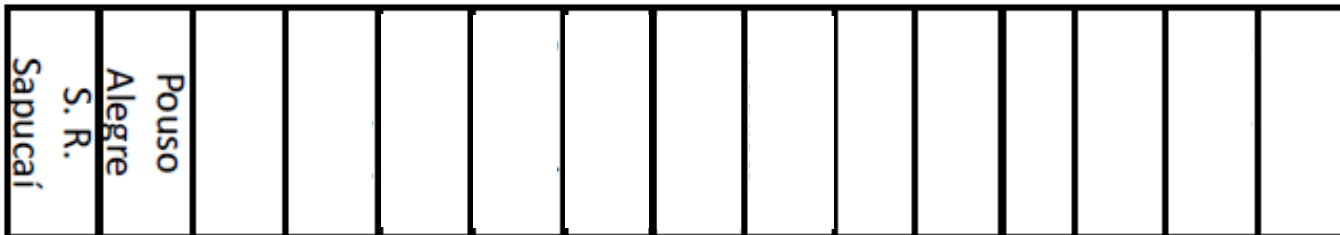
- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



S. R. Sapucaí																				
---------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- Busca Gulosa

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas

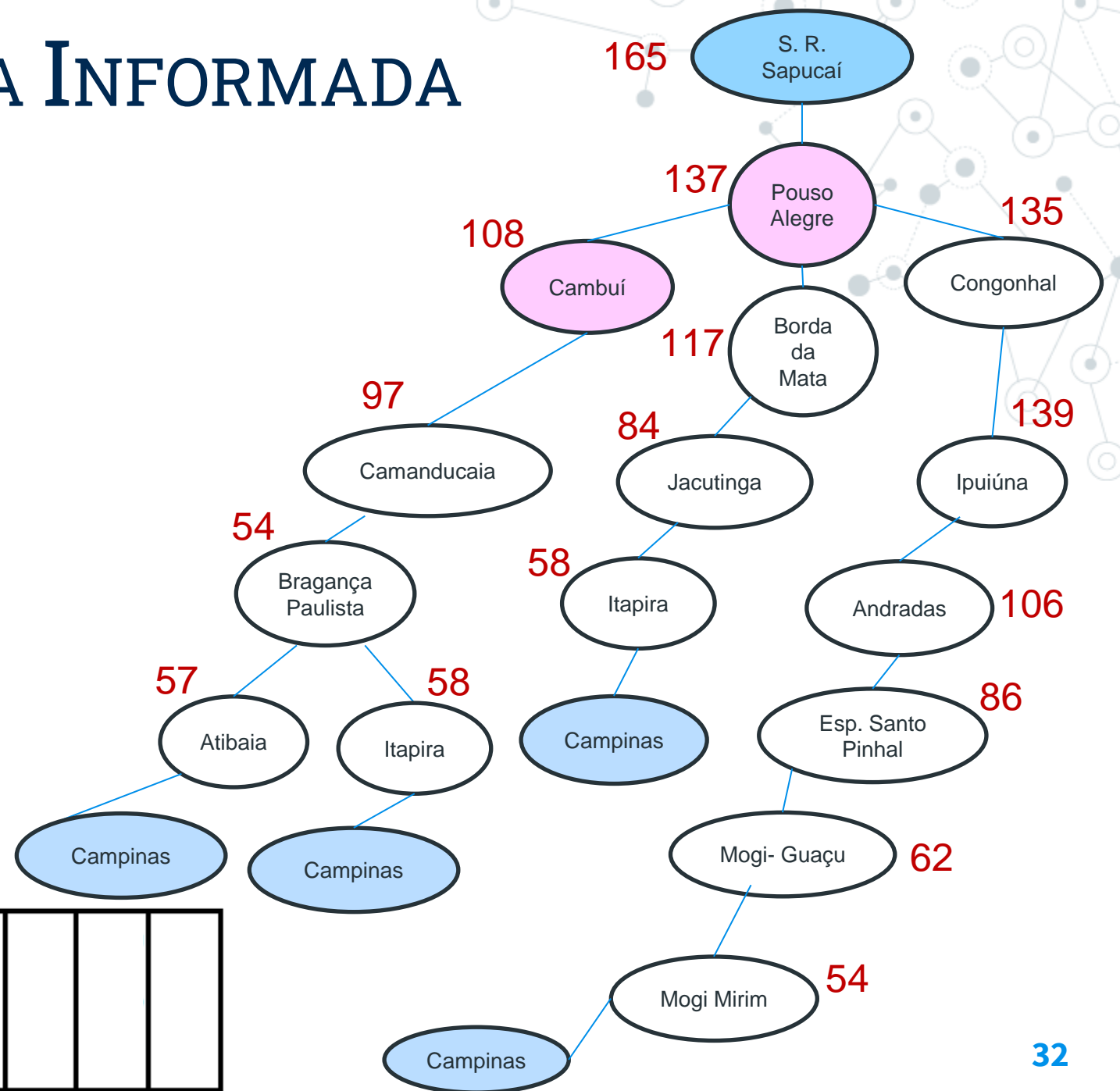


ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



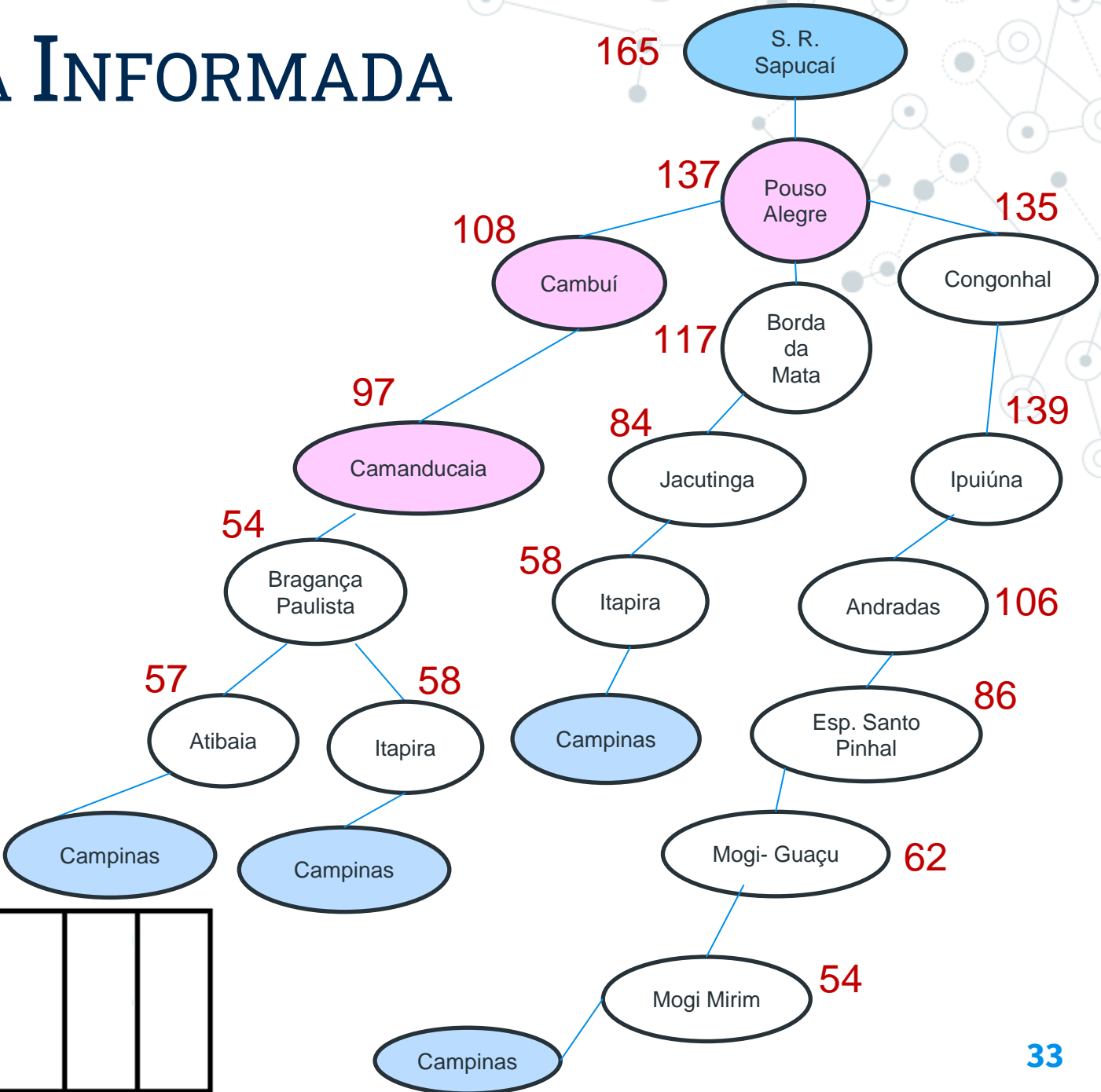
Sapucaí	S. R.	Alegre	Pouso	Cambuí															
---------	-------	--------	-------	--------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



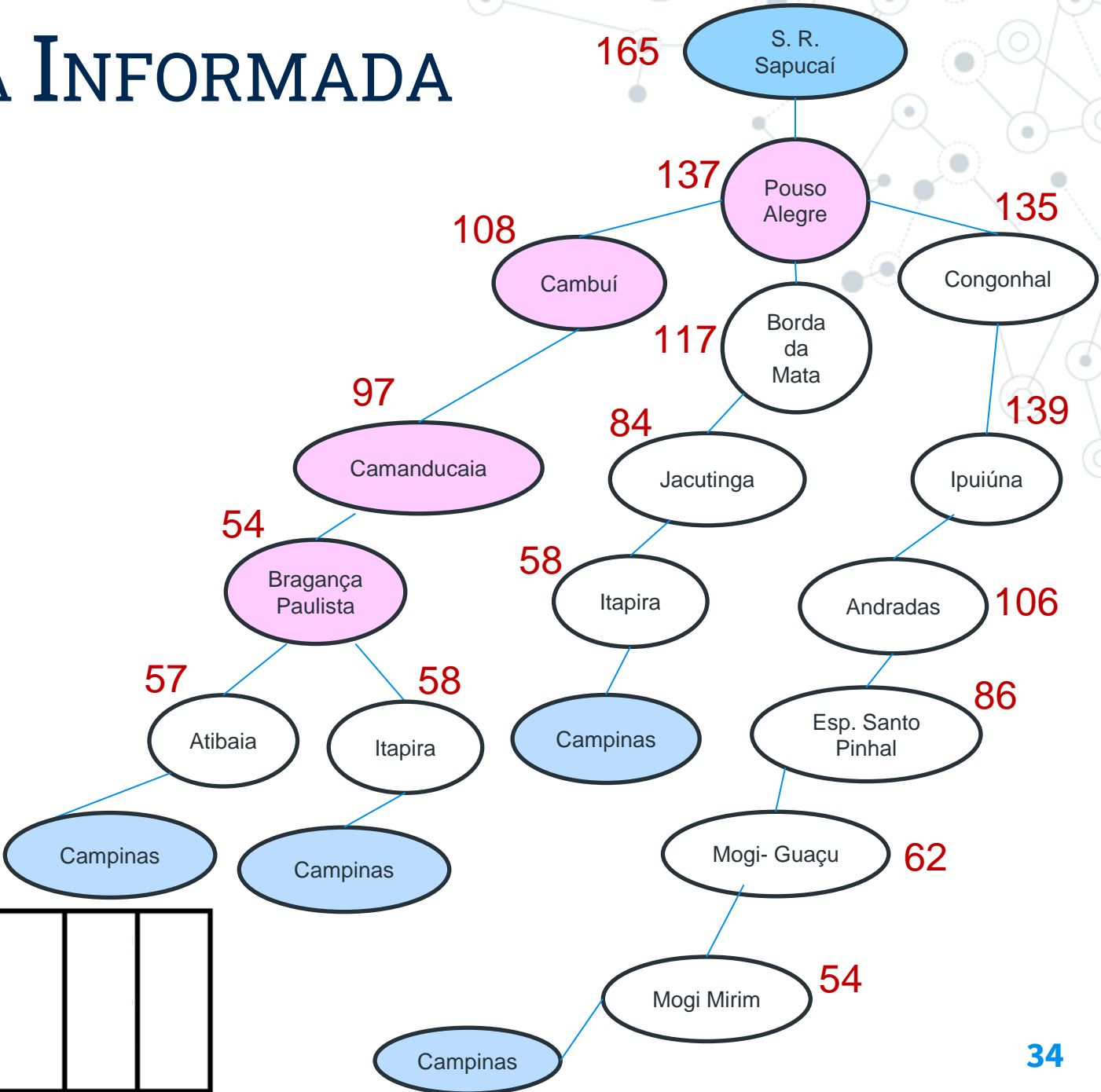
Sapucaí	S. R.	Alegre	Pouso	Cambuí	Camandu														
caia																			

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



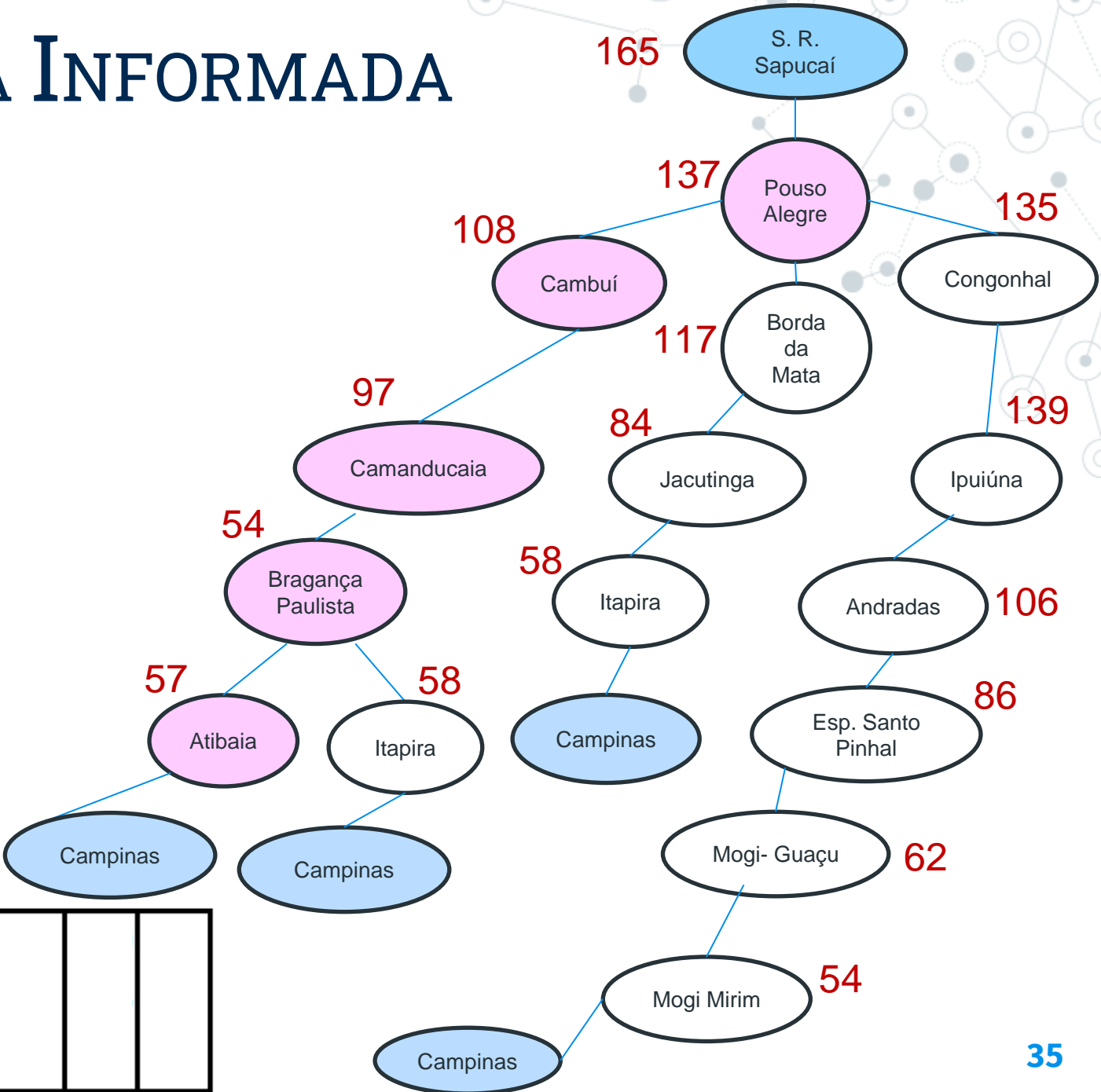
Sapucaí	S. R.	Pouso	Cambuí	Camandu	Bragança														
caia	Paulista																		

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



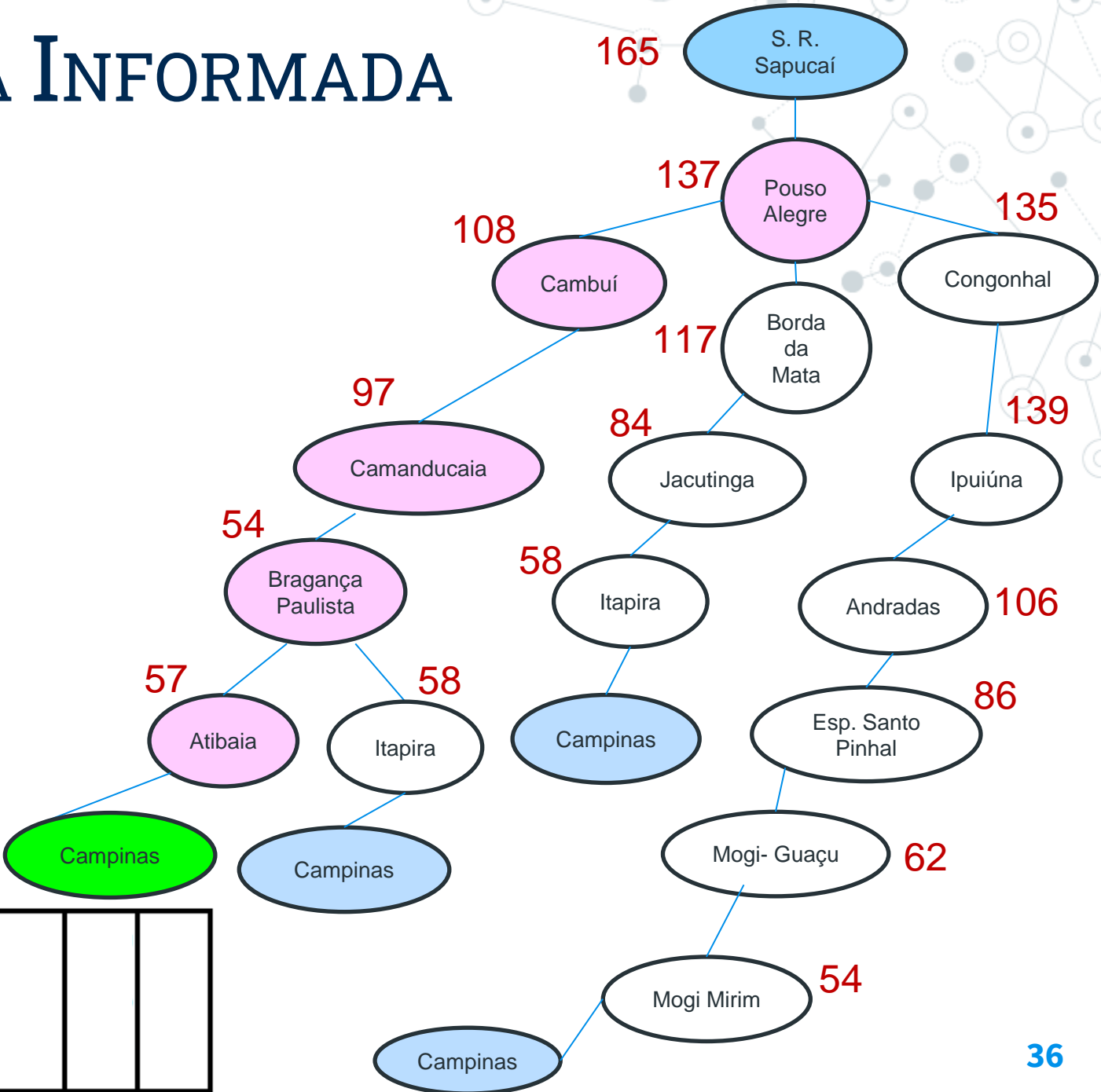
Sapucaí	S. R.	Pouso	Cambuí	Camandu	Bragança	Atibaia													
caia	Paulista	caia																	

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca Gulosa

Problema de Roteamento:

- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



Sapucaí	S. R.	Alegre	Pouso	Cambuí	Camandu	Bragança	Atibaia	Campinas											
caia	Paulista																		

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A* → Minimização do Custo Total Estimado da Solução

- Avalia os nós através da combinação de $g(n)$, o custo para alcançar o nó, e $h(n)$, custo para ir do nó ao objetivo:

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

- Visto que $g(n)$ dá o custo do caminho desde o nó inicial até o nó n e $h(n)$ é o custo estimado do caminho de menor custo de n até o objetivo, temos

$$f(n) = \text{Custo estimado da solução de } \mathbf{MENOR} \text{ custo através de } n$$

A Busca A* é **COMPLETA** e **ÓTIMA**, em certas condições.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A*

○ **Exemplo:** Roteamento na Romênia.

- **Objetivo:** Buscareste
- **Estado Inicial:** Arad
- **Função Heurística:** Distância em linha reta (DLR) definida por h_{DLR} .
- É necessário conhecer as distâncias em linha reta para Bucareste.

Tabela 1 - h_{DLR} para todas as cidades

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

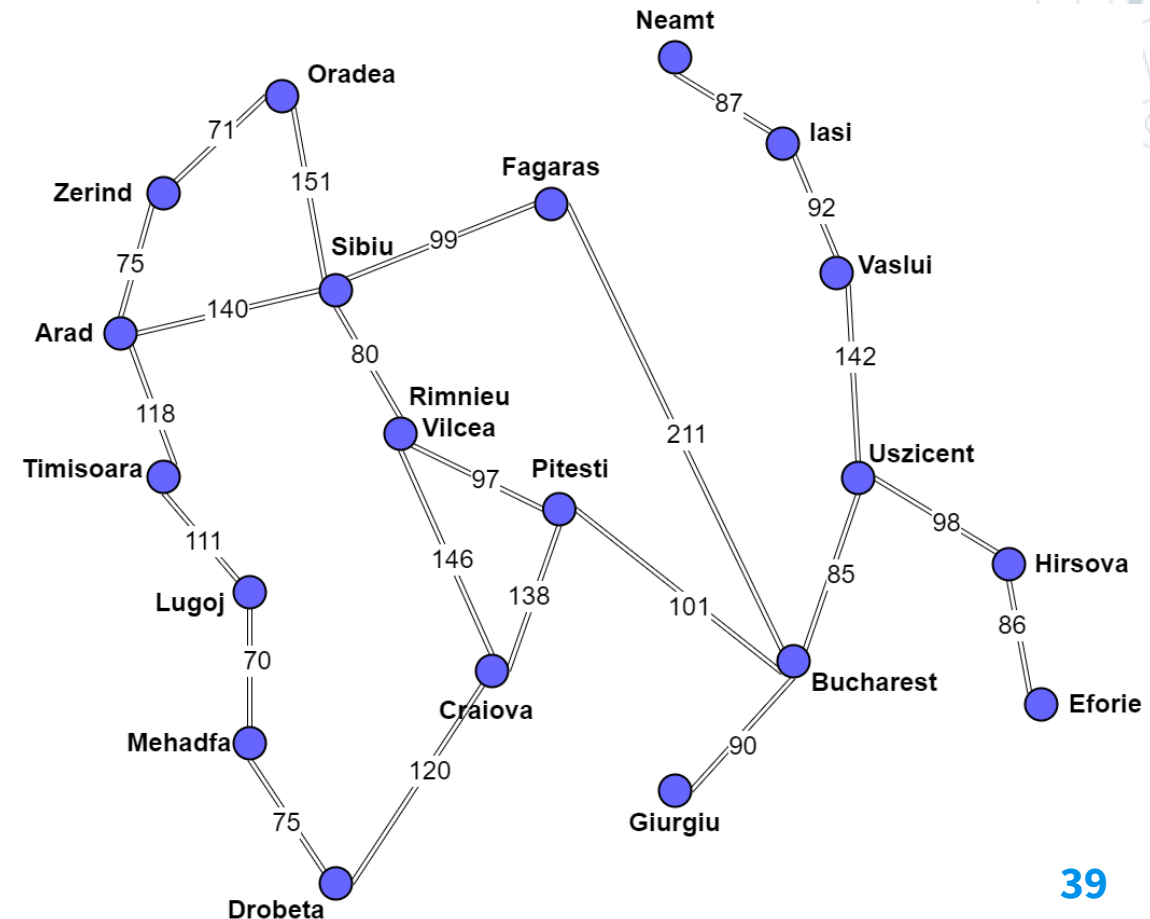
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca A*

- Exemplo:** Roteamento na Romênia.

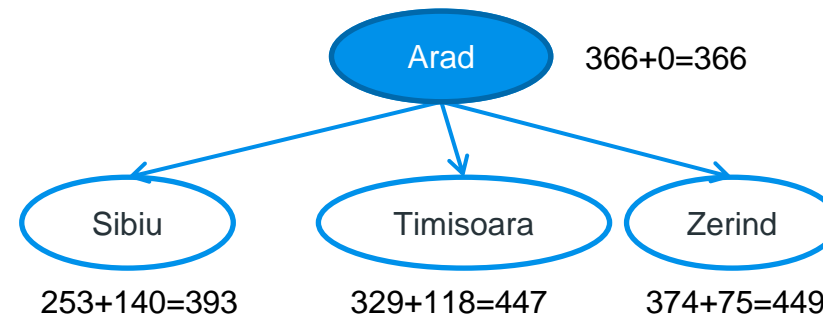
Tabela 1 - h_{DLR} para todas as cidades

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



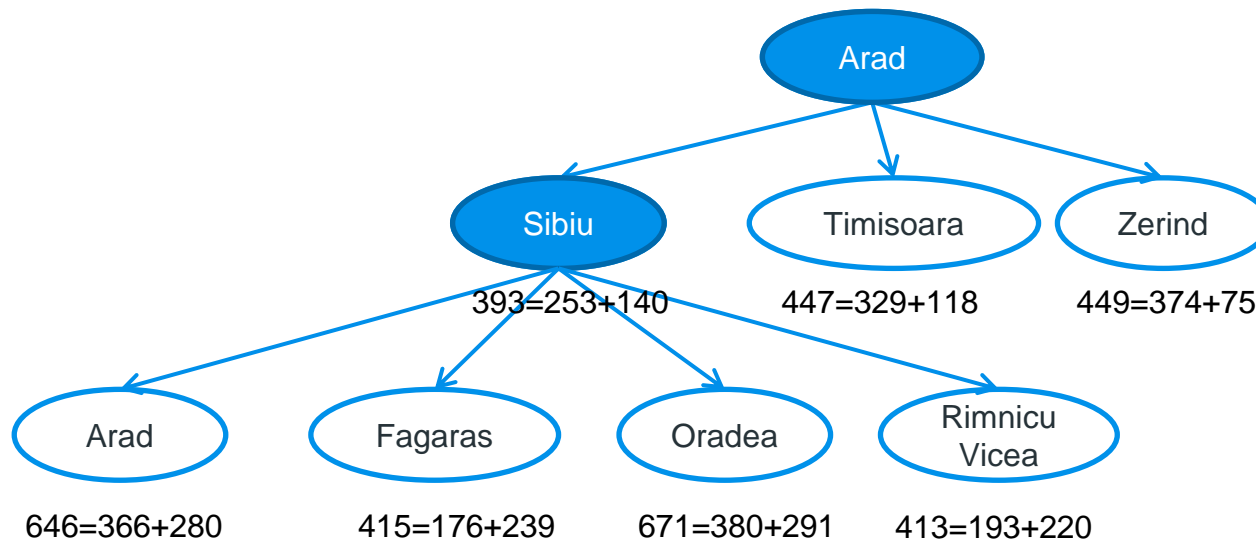
⦿ Busca A*

- **Exemplo:** Roteamento na Romênia.



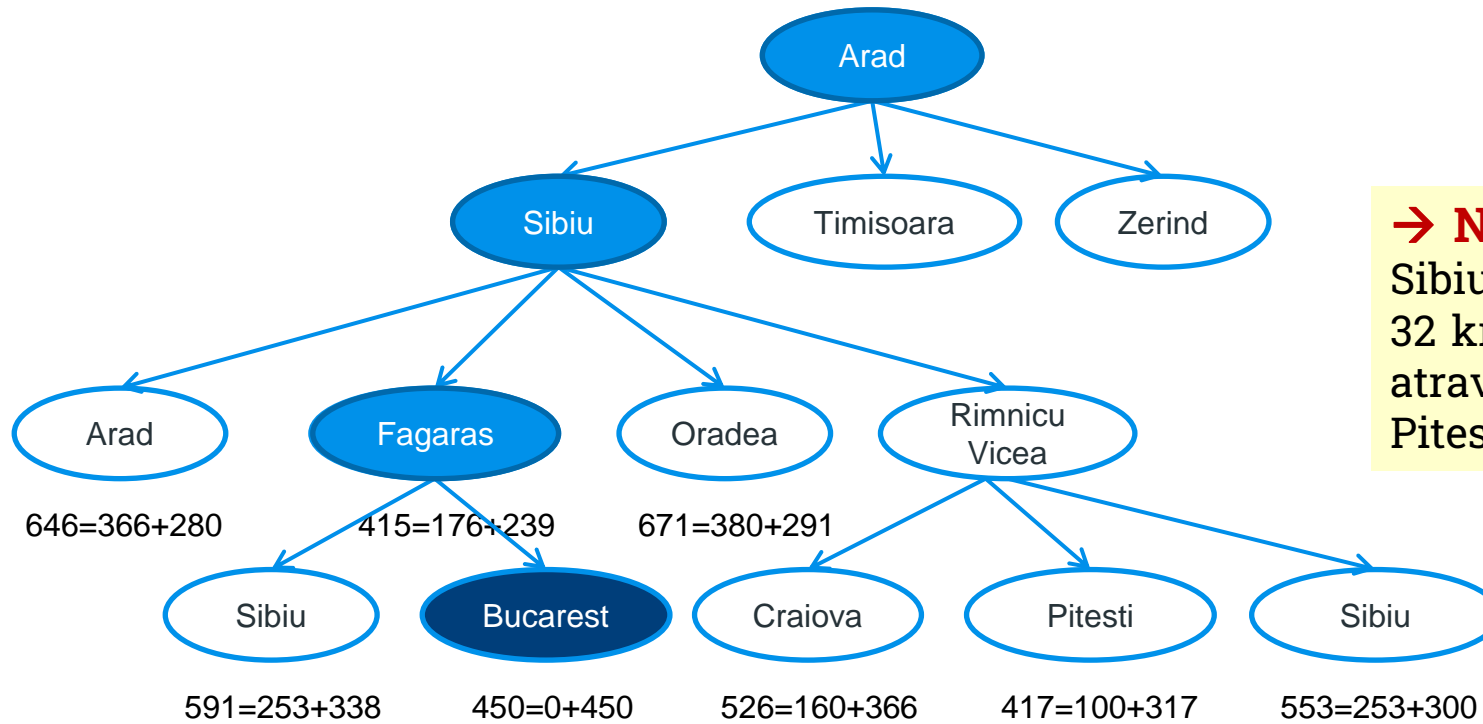
Busca A*

- Exemplo:** Roteamento na Romênia.



Busca A*

- Exemplo:** Roteamento na Romênia.



→ **NÃO é Ótimo:** O caminho via Sibiu e Fagaras para Bucarest é 32 km mais longo que o caminho através de Rimnicu Vilcea e Pitesti.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- ◉ Busca A*
 - **Exemplo:** Roteamento na Romênia.
 - **Objetivo:** Buscareste
 - **Estado Inicial:** Arad
 - **Função Heurística:** Distância em linha reta (DLR) definida por h_{DLR} + Custo de cada caminho.
 - É necessário conhecer as distâncias em linha reta para Bucareste.

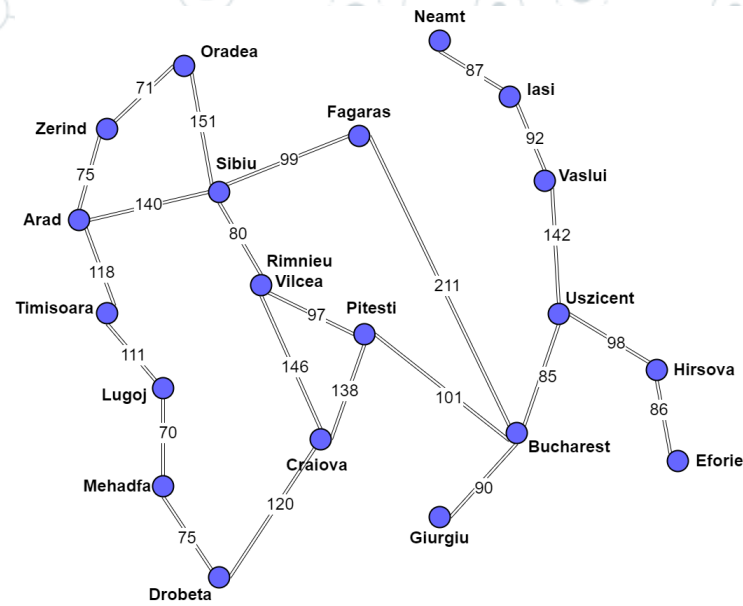


Tabela 1 - h_{DLR} para todas as cidades

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

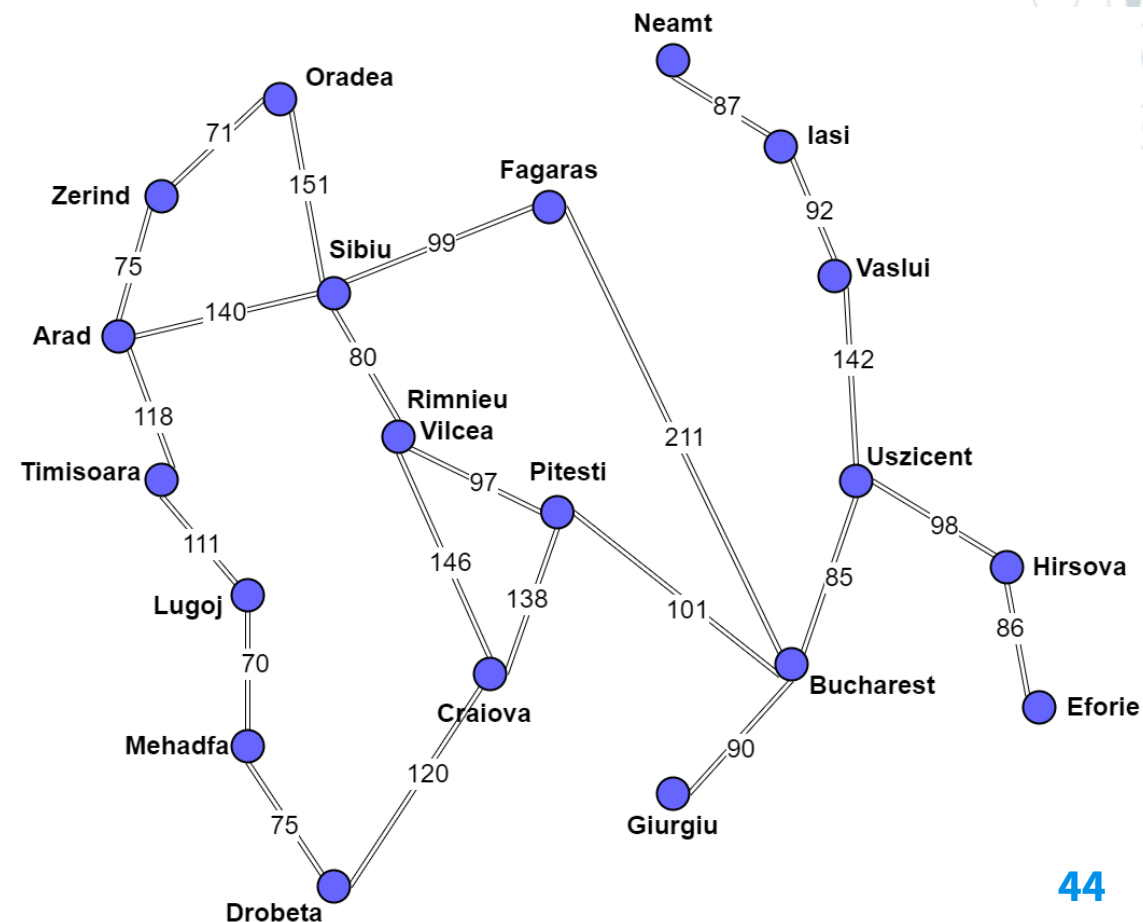
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A*

- **Exemplo:** Roteamento na Romênia.

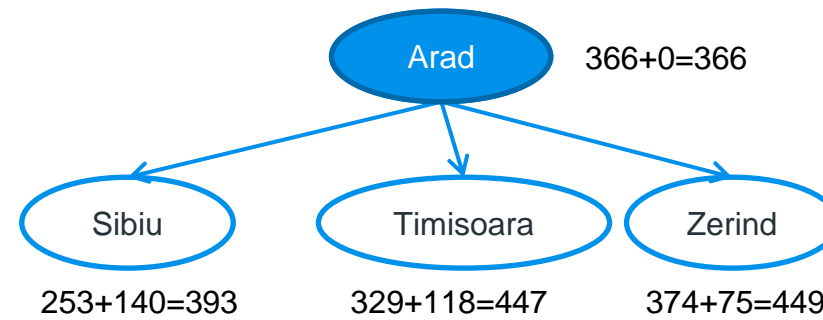
Tabela 1 - h_{DLR} para todas as cidades

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



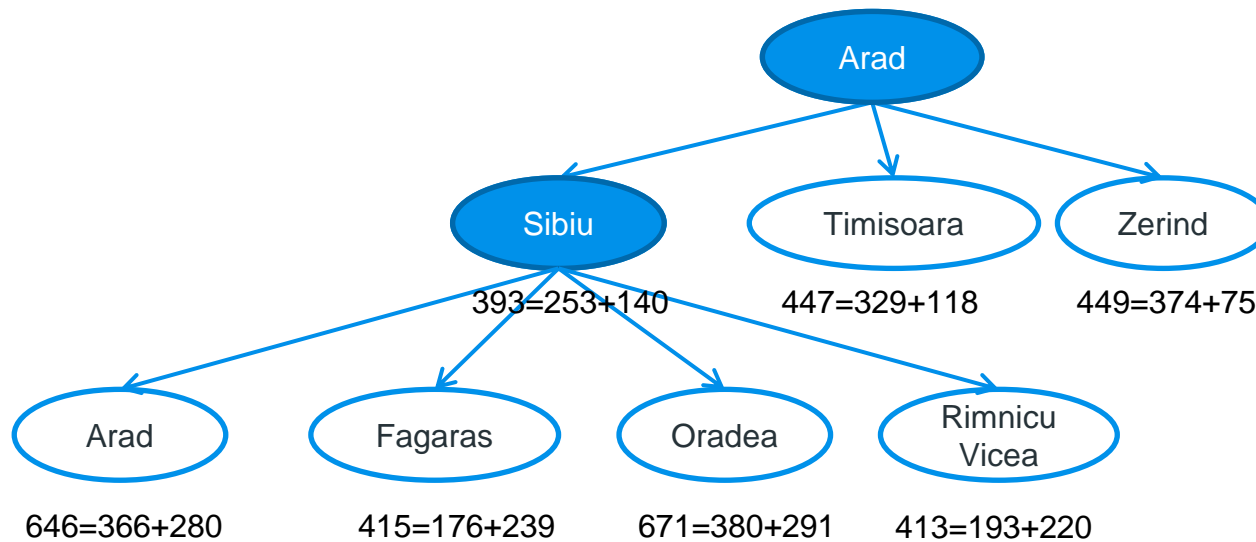
⦿ Busca A*

- **Exemplo:** Roteamento na Romênia.



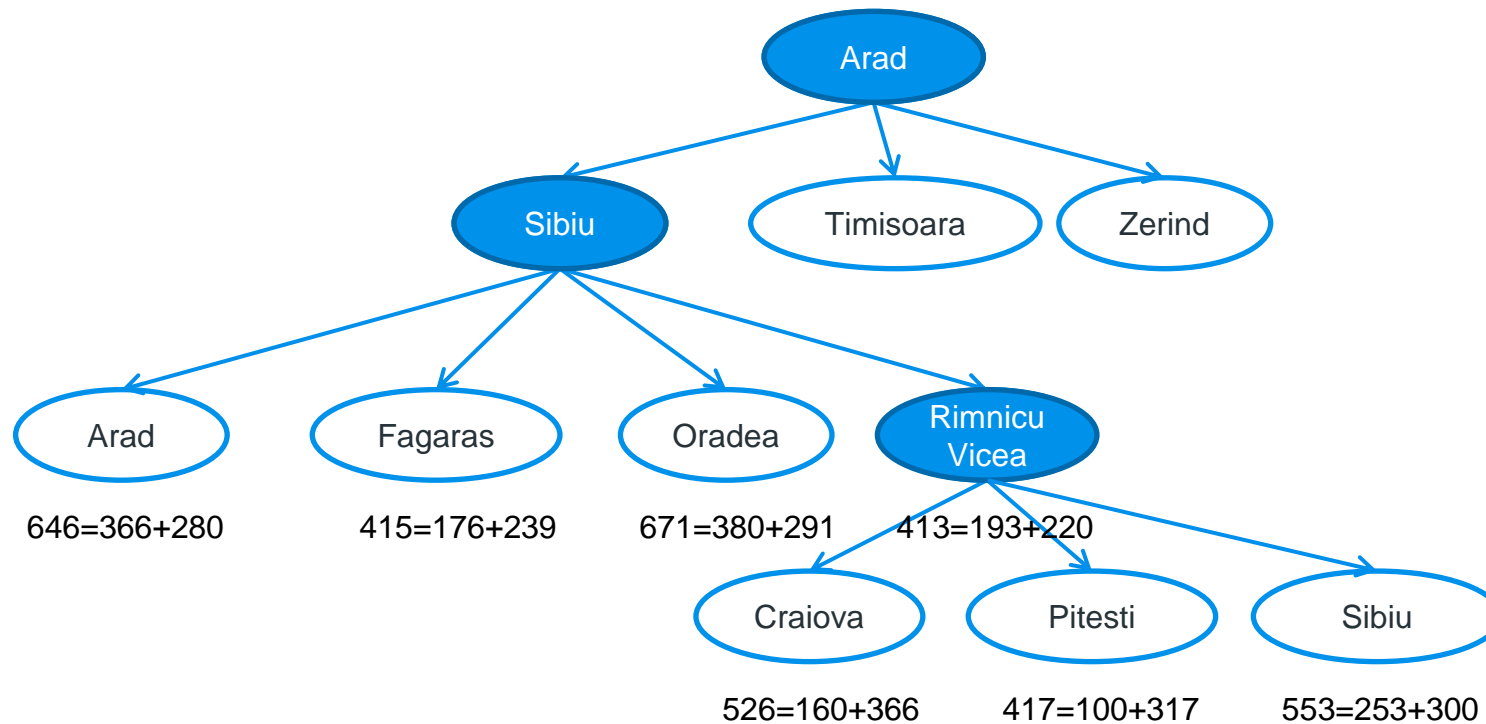
Busca A*

- Exemplo:** Roteamento na Romênia.



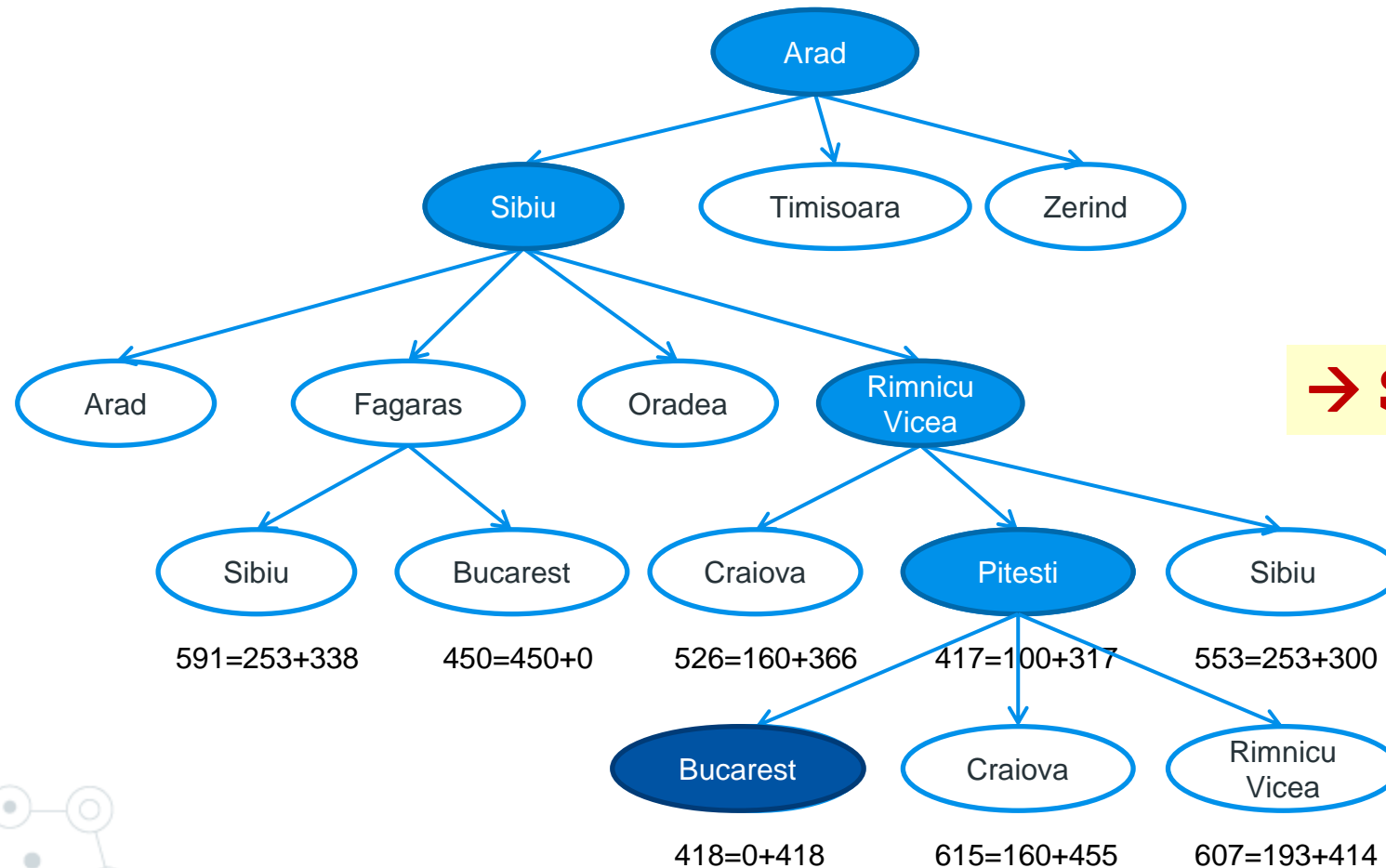
Busca A*

- Exemplo:** Roteamento na Romênia.



Busca A*

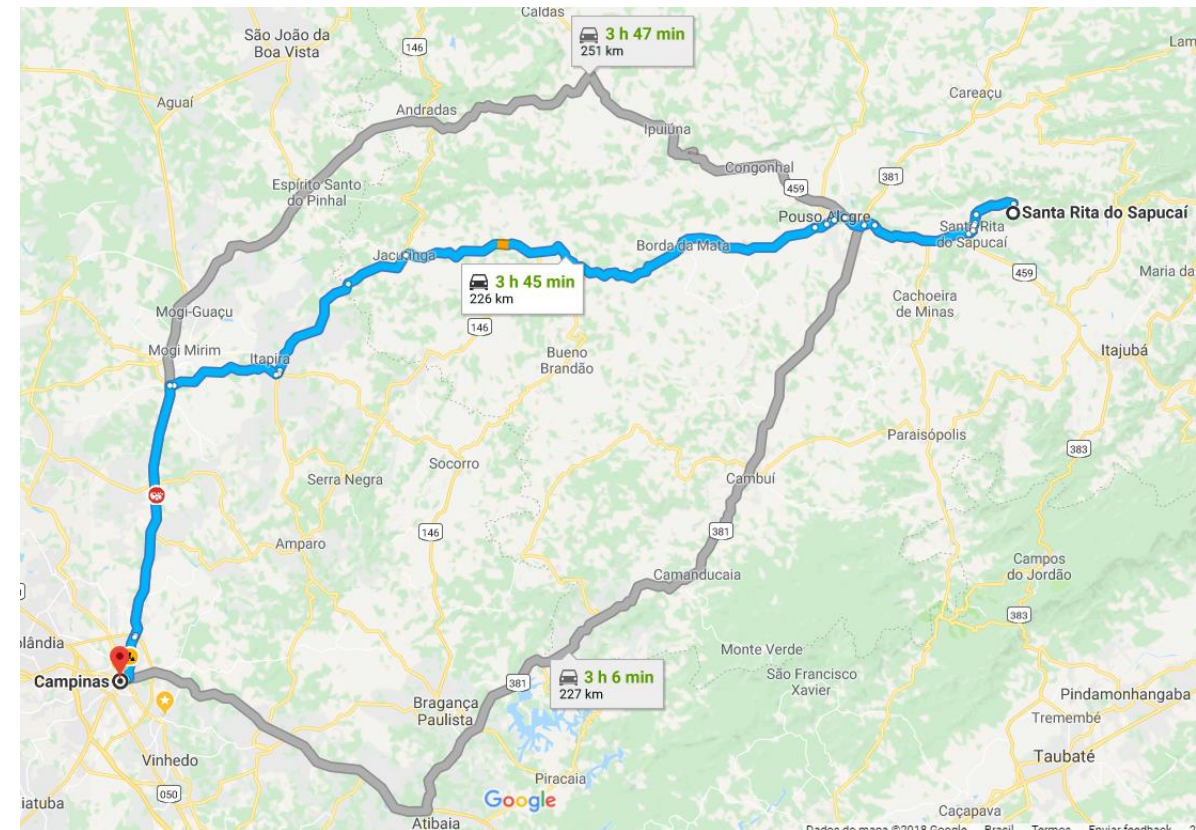
- Exemplo:** Roteamento na Romênia.



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A*

- **Exemplo:** Roteamento Santa Rita do Sapucaí -Campinas.



ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A*

- **Exemplo:** Roteamento Santa Rita do Sapucaí -Campinas.
- **Objetivo:** Campinas
- **Função Heurística:** Distância em linha reta (DLR) definida por h_{DLR} + Custo real que interliga todas as cidades do mapa.

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

◎ Busca A*

- **Exemplo:** Roteamento Santa Rita do Sapucaí - Campinas.

Tabela 1 - h_{DLR} para todas as cidades

S. R. Sapucaí	165	Jacutinga	84
Pouso Alegre	137	Itapira	58
Cambuí	108	Congonhal	135
Camanducaia	97	Ipuiúna	139
Bragança Paulista	54	Andradas	106
Atibaia	57	Esp. Santo Pinhal	86
Campinas	0	Mogi-Guaçu	62
Borda da Mata	117	Mogi Mirim	54

Fonte: <https://www.adistanciaentre.com>

Tabela 2 – Distância real entre as cidades

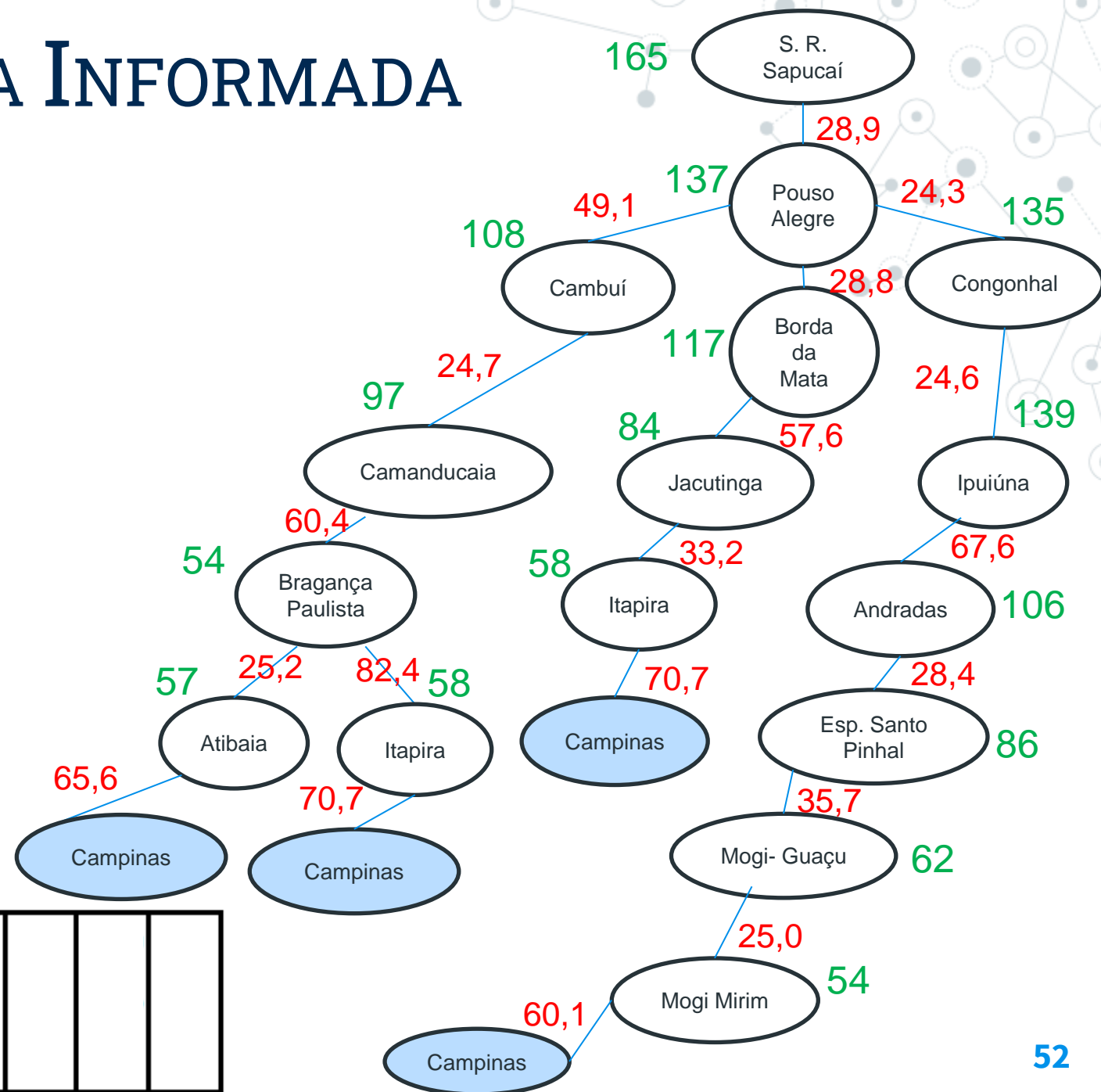
S. R. Sapucaí	Pouso Alegre	28,9
Pouso Alegre	Cambuí	49,1
Pouso Alegre	Congonhal	24,3
Pouso Alegre	Borda da Mata	28,8
Cambuí	Camanducaia	24,7
Camanducaia	Bragança Paulista	60,4
Bragança Paulista	Atibaia	25,2
Bragança Paulista	Itapira	82,4
Atibaia	Campinas	65,6
Itapira	Campinas	70,7
Borda da Mata	Jacutinga	57,6
Jacutinga	Itapira	33,2
Congonhal	Ipuiúna	24,6
Ipuiúna	Andradas	67,5
Andradas	Esp. Santo Pinhal	28,4
Esp. Santo Pinhal	Mogi-Guaçu	35,7
Mogi Mirim	Campinas	60,1

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca A*

Problema de Roteamento:

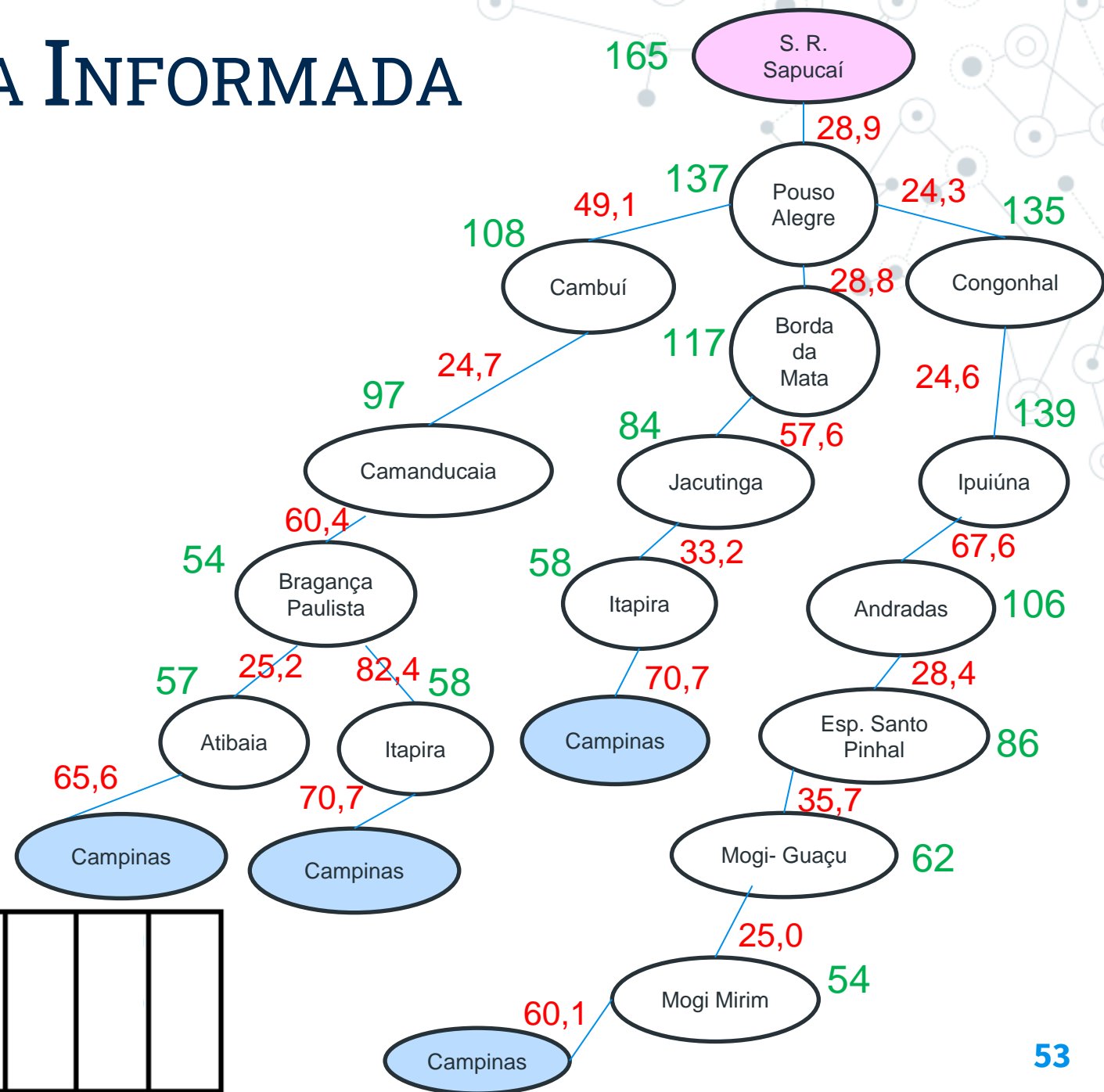
- **Origem:** Santa Rita do Sapucaí
- **Destino:** Campinas



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

○ Busca A*

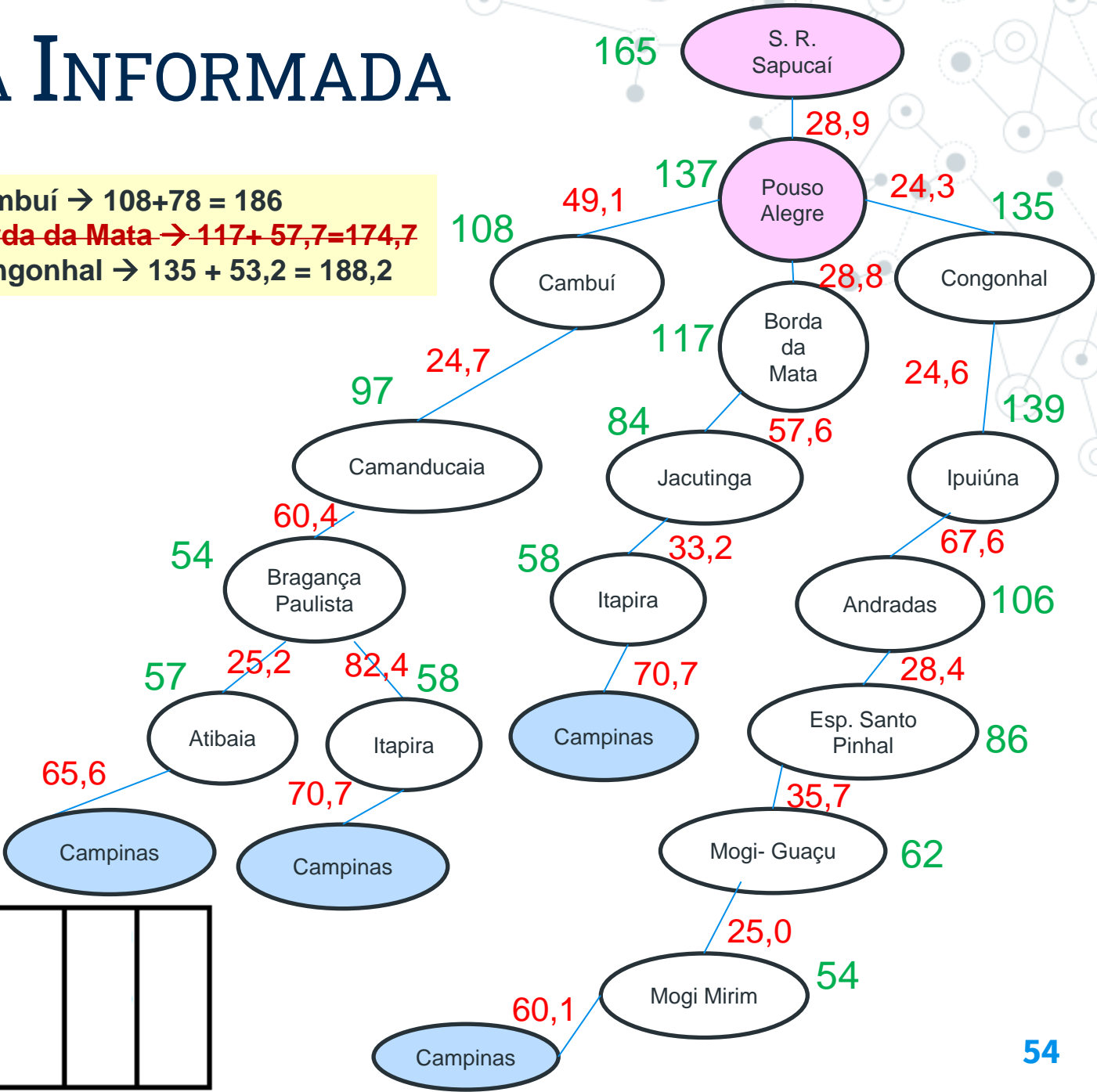


S. R. Sapucaí																			
------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca A*

Cambuí → 108+78 = 186
~~Borda da Mata~~ → ~~117~~+ 57,7=174,7
Congonhal → 135 + 53,2 = 188,2



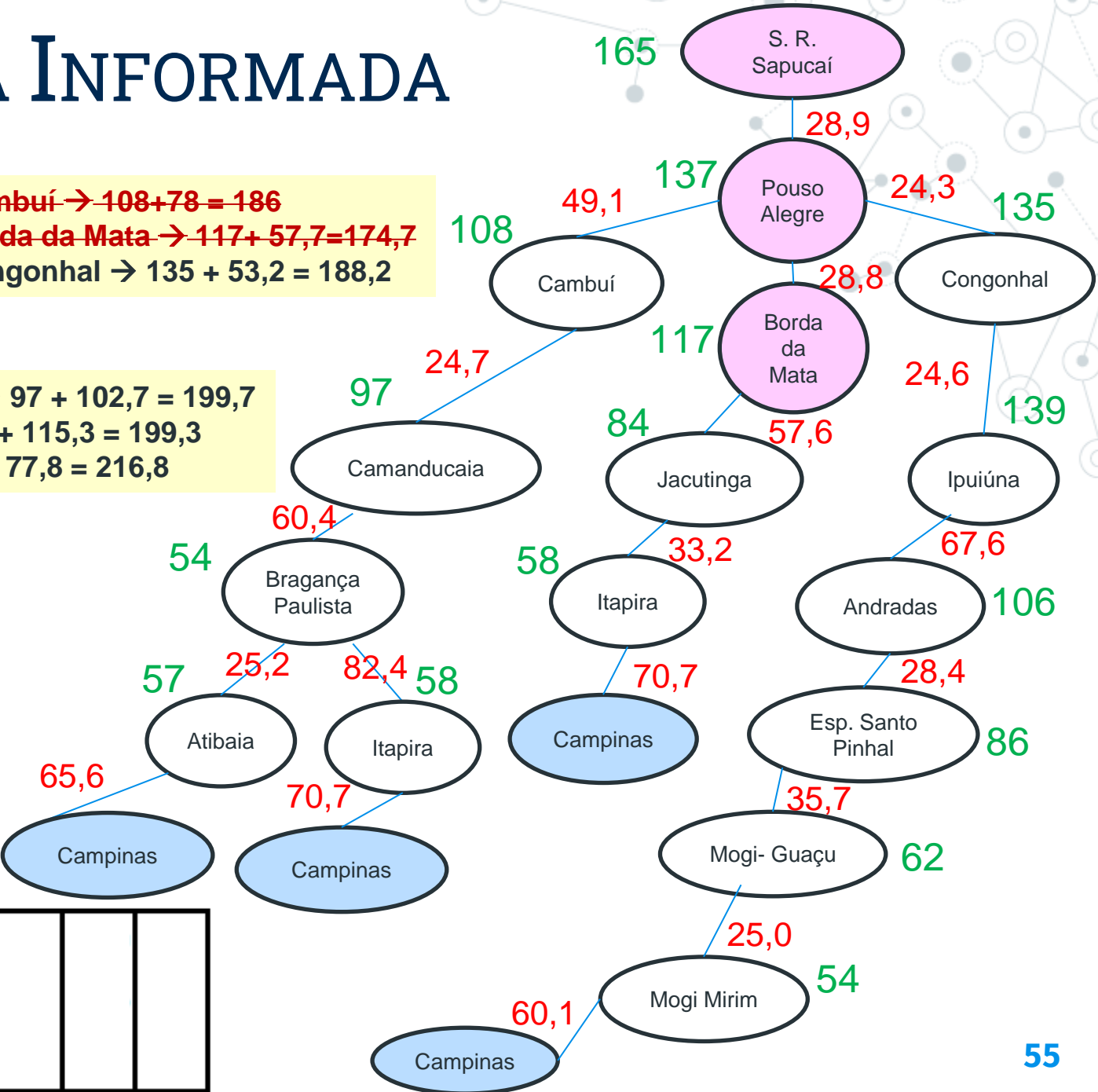
Sapucaí	Pouso Alegre																		
S. R.																			

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Busca A*

~~Cambuí~~ → ~~108~~ + 78 = 186
~~Borda da Mata~~ → ~~117~~ + 57,7 = 174,7
Congonhal → 135 + 53,2 = 188,2

Camanducaia → 97 + 102,7 = 199,7
Jacutinga → 84 + 115,3 = 199,3
Ipuiúna → 139 + 77,8 = 216,8



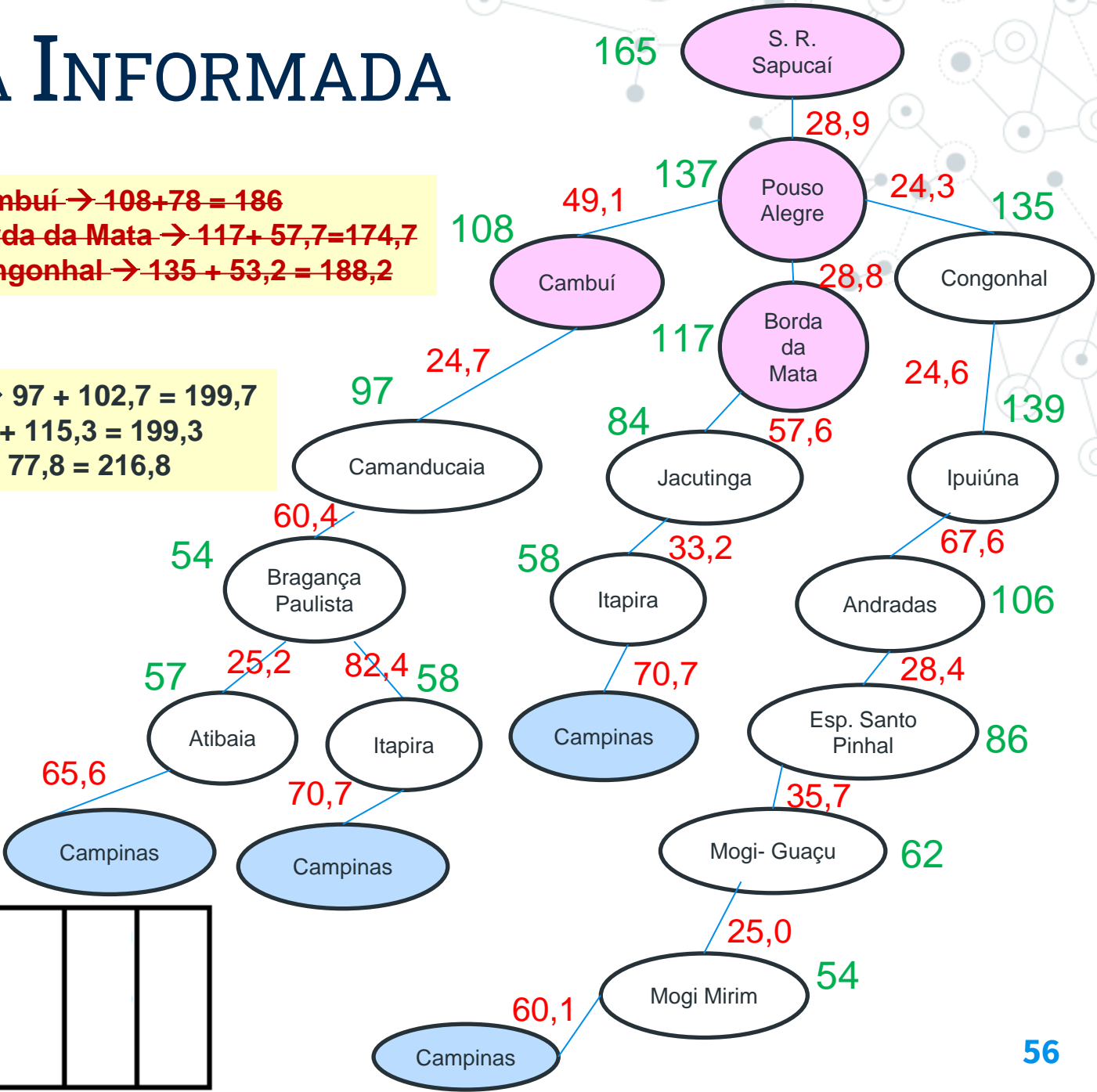
Sapucaí	S. R.	Pouso Alegre	Borda da Mata																
---------	-------	--------------	---------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Busca A*

~~Cambuí~~ → ~~108+78=186~~
~~Borda da Mata~~ → ~~117+57,7=174,7~~
~~Congonhal~~ → ~~135+53,2=188,2~~

Camanducaia → 97 + 102,7 = 199,7
Jacutinga → 84 + 115,3 = 199,3
Ipuiúna → 139 + 77,8 = 216,8



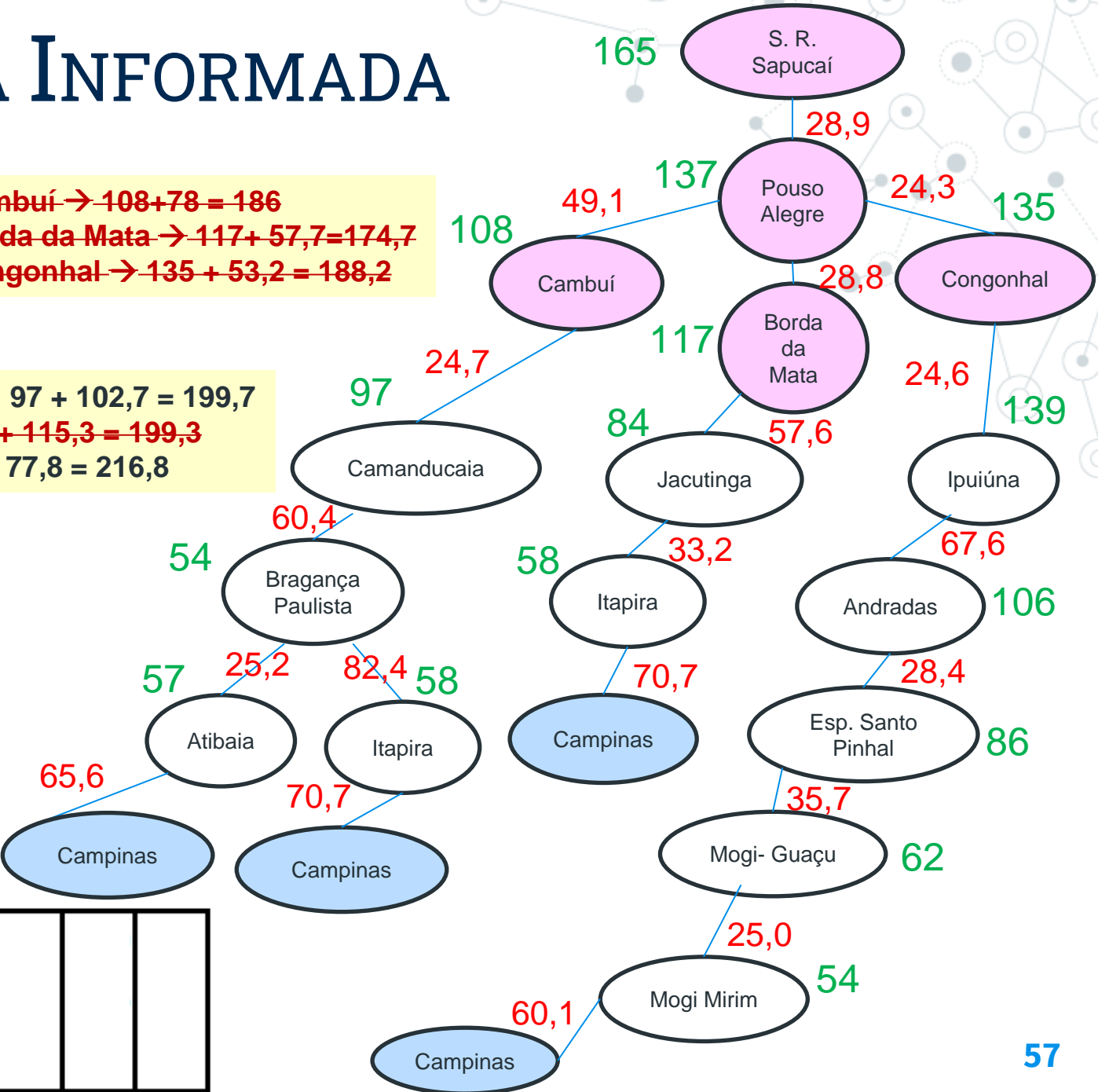
Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí															
---------	-------	--------------	---------------	--------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Busca A*

~~Cambuí~~ → ~~108+78=186~~
~~Borda da Mata~~ → ~~117+ 57,7=174,7~~
~~Congonhal~~ → ~~135 + 53,2 = 188,2~~

Camanducaia → 97 + 102,7 = 199,7
Jacutinga → 84 + 115,3 = 199,3
Ipuiúna → 139 + 77,8 = 216,8



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal														
---------	-------	--------------	---------------	--------	-----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--

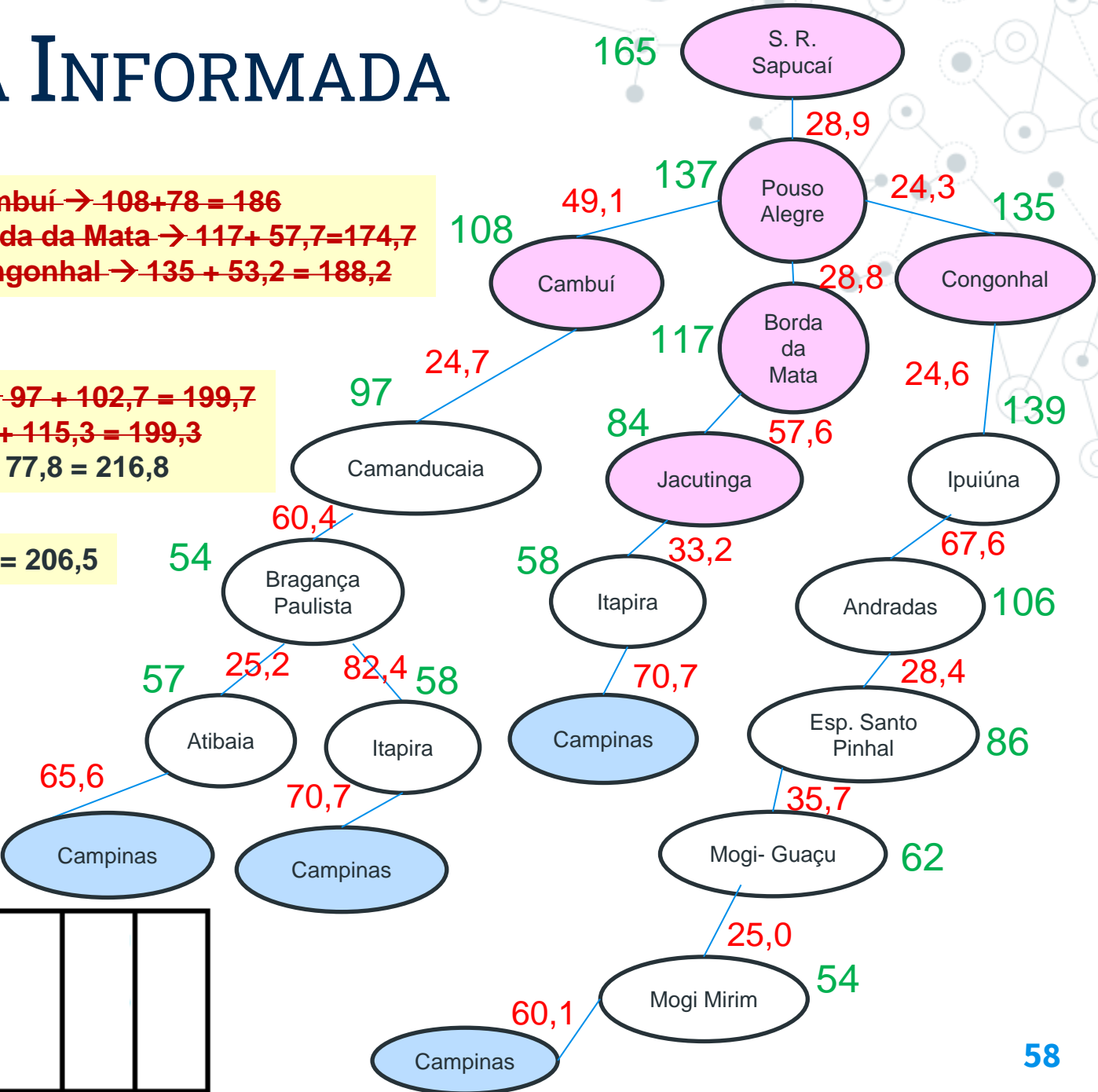
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Busca A*

Cambuí → $108 + 78 = 186$
Borda da Mata → $117 + 57,7 = 174,7$
Congonhal → $135 + 53,2 = 188,2$

Camanducaia → $97 + 102,7 = 199,7$
Jacutinga → $84 + 115,3 = 199,3$
Ipuiúna → $139 + 77,8 = 216,8$

Itapira → $58 + 148,5 = 206,5$



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga													
---------	-------	--------------	---------------	--------	-----------	-----------	--	--	--	--	--	--	--	--	--	--	--	--	--

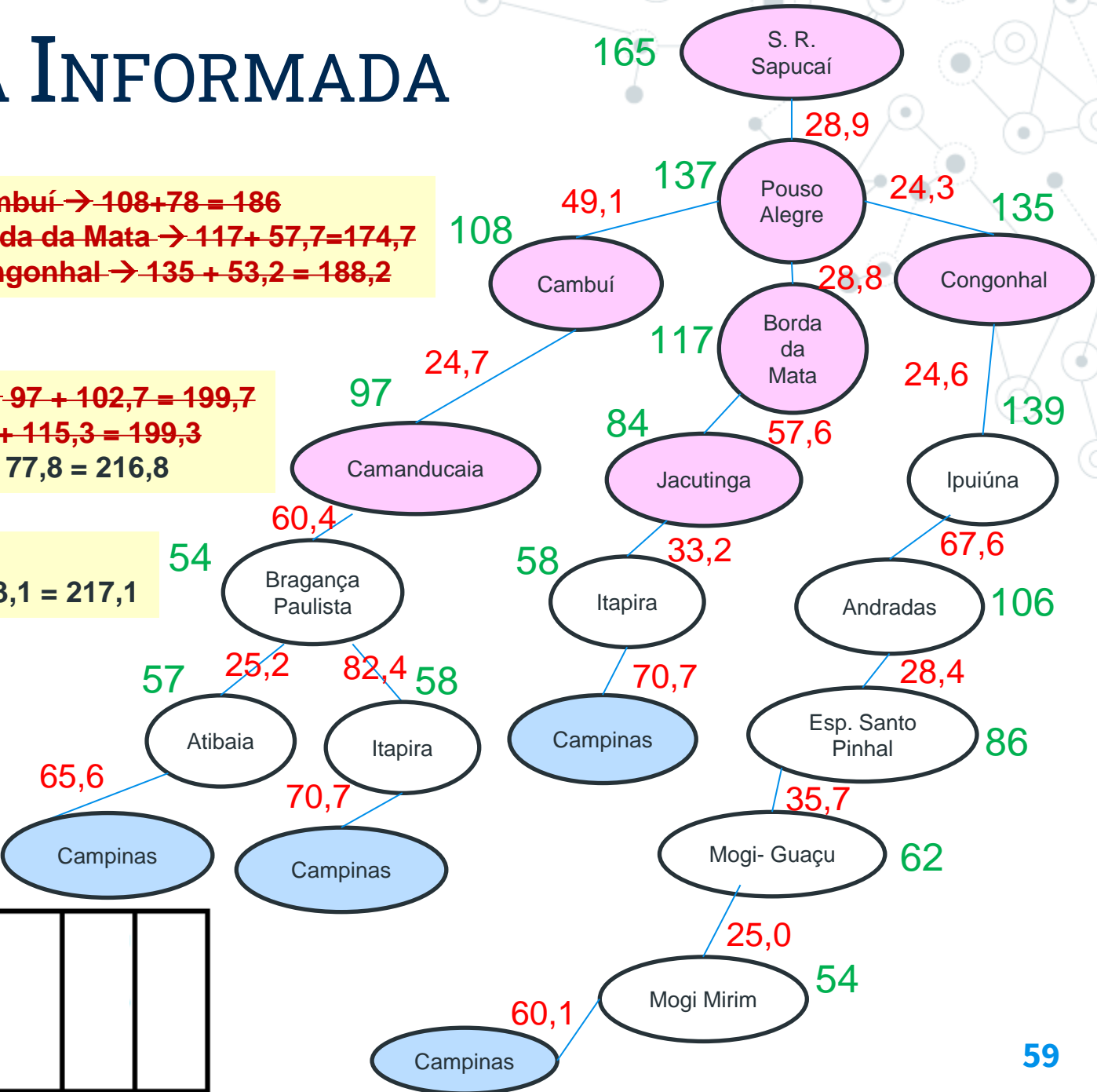
ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

- Busca A*

Cambuí → $108 + 78 = 186$
Borda da Mata → $117 + 57,7 = 174,7$
Congonhal → $135 + 53,2 = 188,2$

Camanducaia → $97 + 102,7 = 199,7$
Jacutinga → $84 + 115,3 = 199,3$
Ipuiúna → $139 + 77,8 = 216,8$

Itapira → $58 + 148,5 = 206,5$
Bragança Paulista → $54 + 163,1 = 217,1$



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga	Camanducaia												
---------	-------	--------------	---------------	--------	-----------	-----------	-------------	--	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

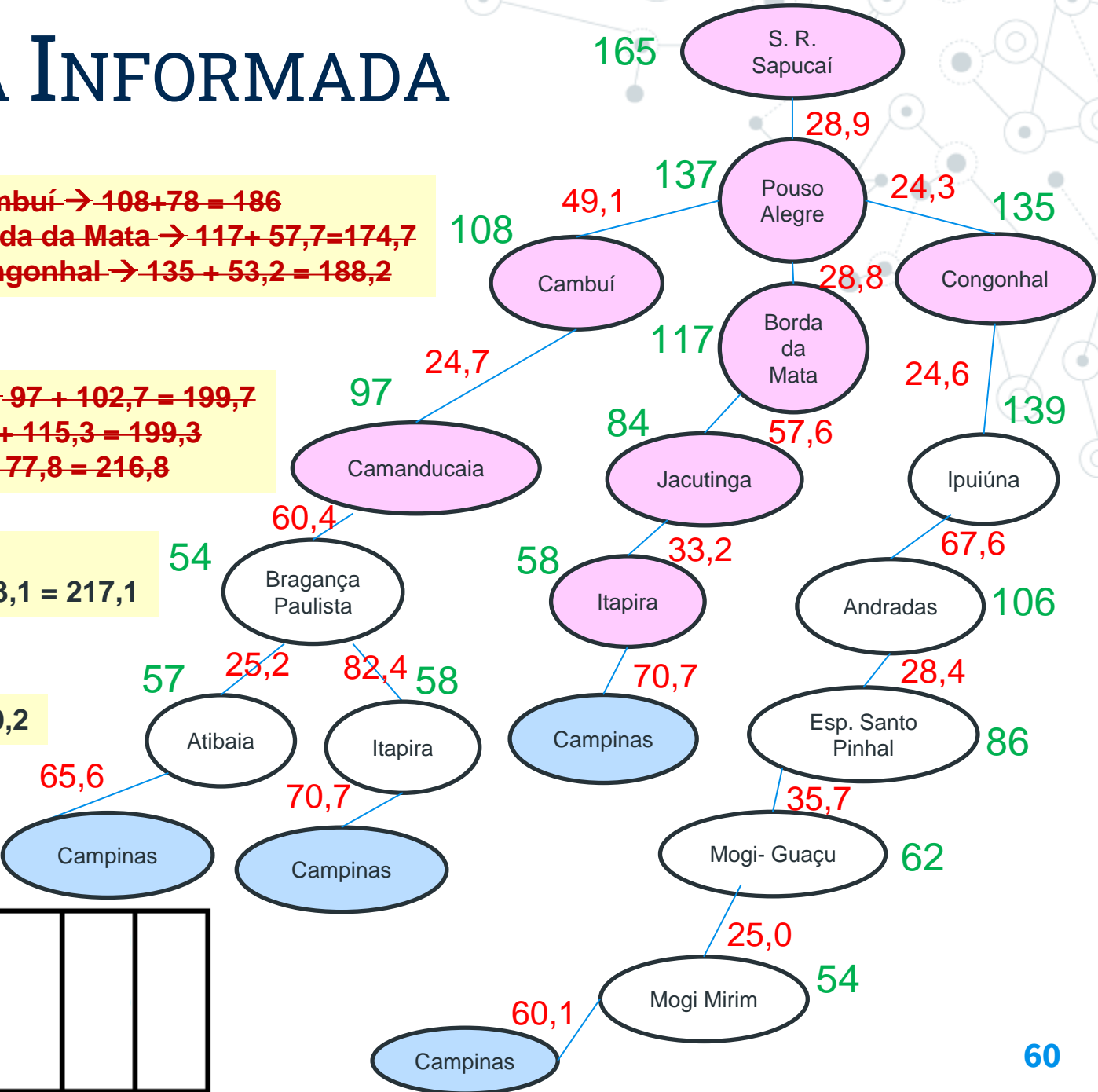
- Busca A*

~~Cambuí~~ → ~~108 + 78 = 186~~
~~Borda da Mata~~ → ~~117 + 57,7 = 174,7~~
~~Congonhal~~ → ~~135 + 53,2 = 188,2~~

~~Camanducaia~~ → ~~97 + 102,7 = 199,7~~
~~Jacutinga~~ → ~~84 + 115,3 = 199,3~~
~~Ipuiúna~~ → ~~139 + 77,8 = 216,8~~

~~Itapira~~ → ~~58 + 148,5 = 206,5~~
~~Bragança Paulista~~ → ~~54 + 163,1 = 217,1~~

~~Campinas~~ → ~~0 + 219,2 = 219,2~~



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga	Camanducaia	Itapira											
---------	-------	--------------	---------------	--------	-----------	-----------	-------------	---------	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

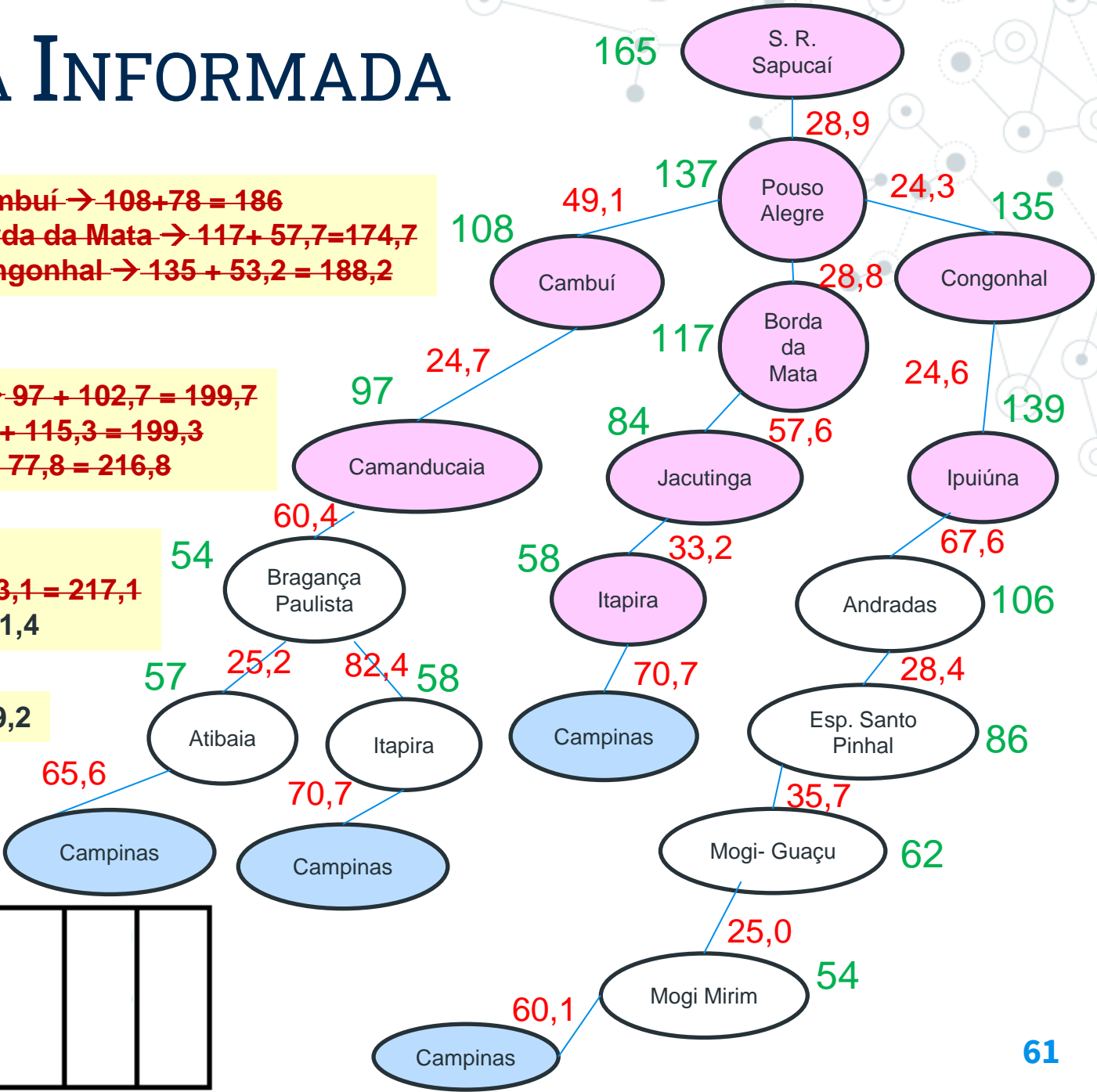
- Busca A*

Cambuí → $108 + 78 = 186$
Borda da Mata → $117 + 57,7 = 174,7$
Congonhal → $135 + 53,2 = 188,2$

Camanducaia → $97 + 102,7 = 199,7$
Jacutinga → $84 + 115,3 = 199,3$
Ipuiúna → $139 + 77,8 = 216,8$

Itapira → $58 + 148,5 = 206,5$
Bragança Paulista → $54 + 163,1 = 217,1$
Andradas → $106 + 145,4 = 251,4$

Campinas → $0 + 219,2 = 219,2$



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga	Camanducaia	Ipuiúna											
---------	-------	--------------	---------------	--------	-----------	-----------	-------------	---------	--	--	--	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

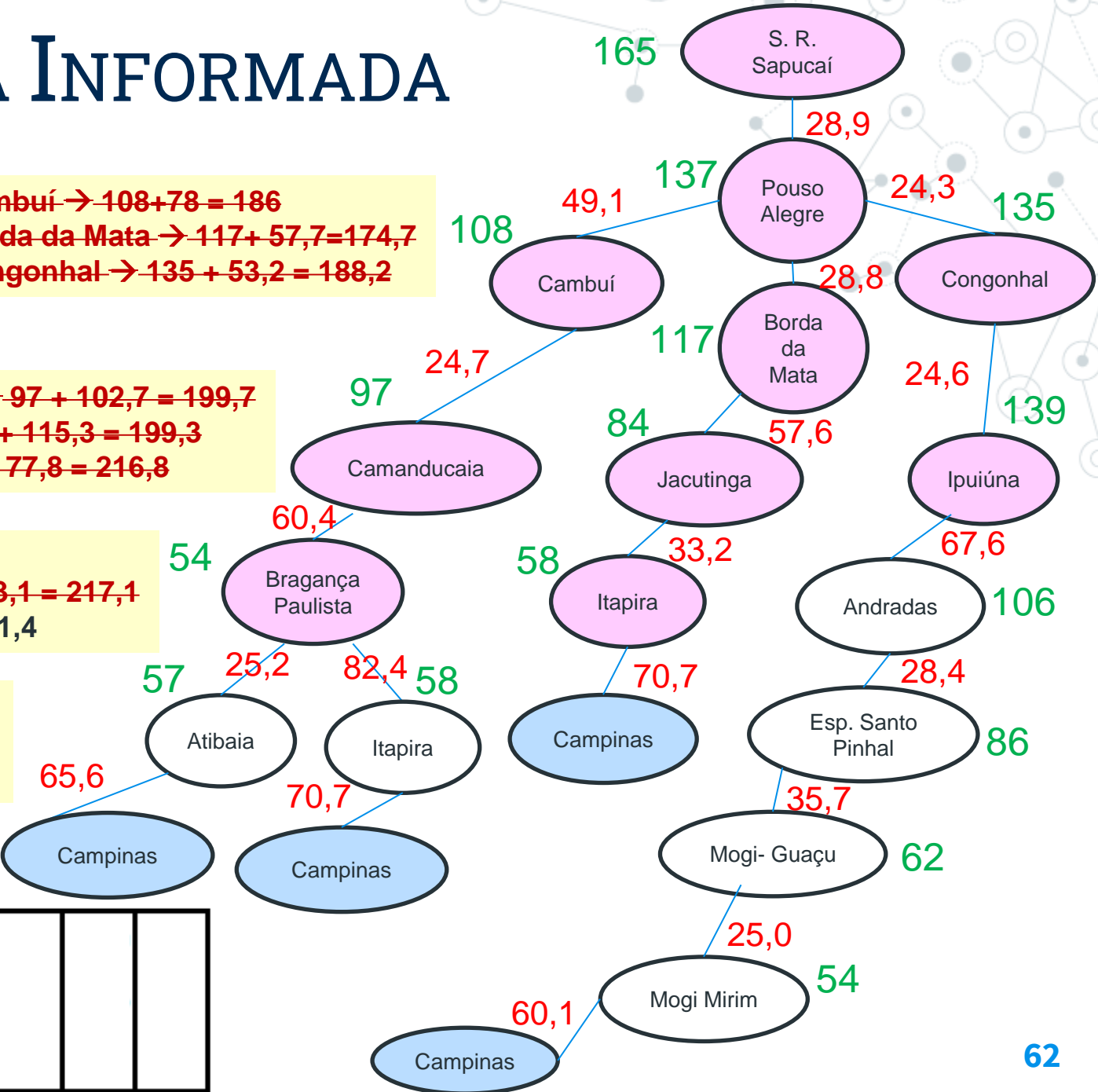
- Busca A*

~~Cambuí~~ $\rightarrow 108 + 78 = 186$
~~Borda da Mata~~ $\rightarrow 117 + 57,7 = 174,7$
~~Congonhal~~ $\rightarrow 135 + 53,2 = 188,2$

~~Camanducaia~~ $\rightarrow 97 + 102,7 = 199,7$
~~Jacutinga~~ $\rightarrow 84 + 115,3 = 199,3$
~~Ipuiúna~~ $\rightarrow 139 + 77,8 = 216,8$

~~Itapira~~ $\rightarrow 58 + 148,5 = 206,5$
~~Bragança Paulista~~ $\rightarrow 54 + 163,1 = 217,1$
~~Andradas~~ $\rightarrow 106 + 145,4 = 251,4$

~~Atibaia~~ $\rightarrow 57 + 188,3 = 245,3$
~~Itapira~~ $\rightarrow 58 + 245,5 = 303,5$
~~Campinas~~ $\rightarrow 0 + 219,2 = 219,2$



Sapucaí	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga	Camanducaia	Ipuiúna	Bragança Paulista								
---------	-------	--------------	---------------	--------	-----------	-----------	-------------	---------	-------------------	--	--	--	--	--	--	--	--

ESTRATÉGIA DE BUSCA INFORMADA (HEURÍSTICA)

Busca A*

~~Cambuí~~ → ~~108 + 78 = 186~~
~~Borda da Mata~~ → ~~117 + 57,7 = 174,7~~
~~Congonhal~~ → ~~135 + 53,2 = 188,2~~

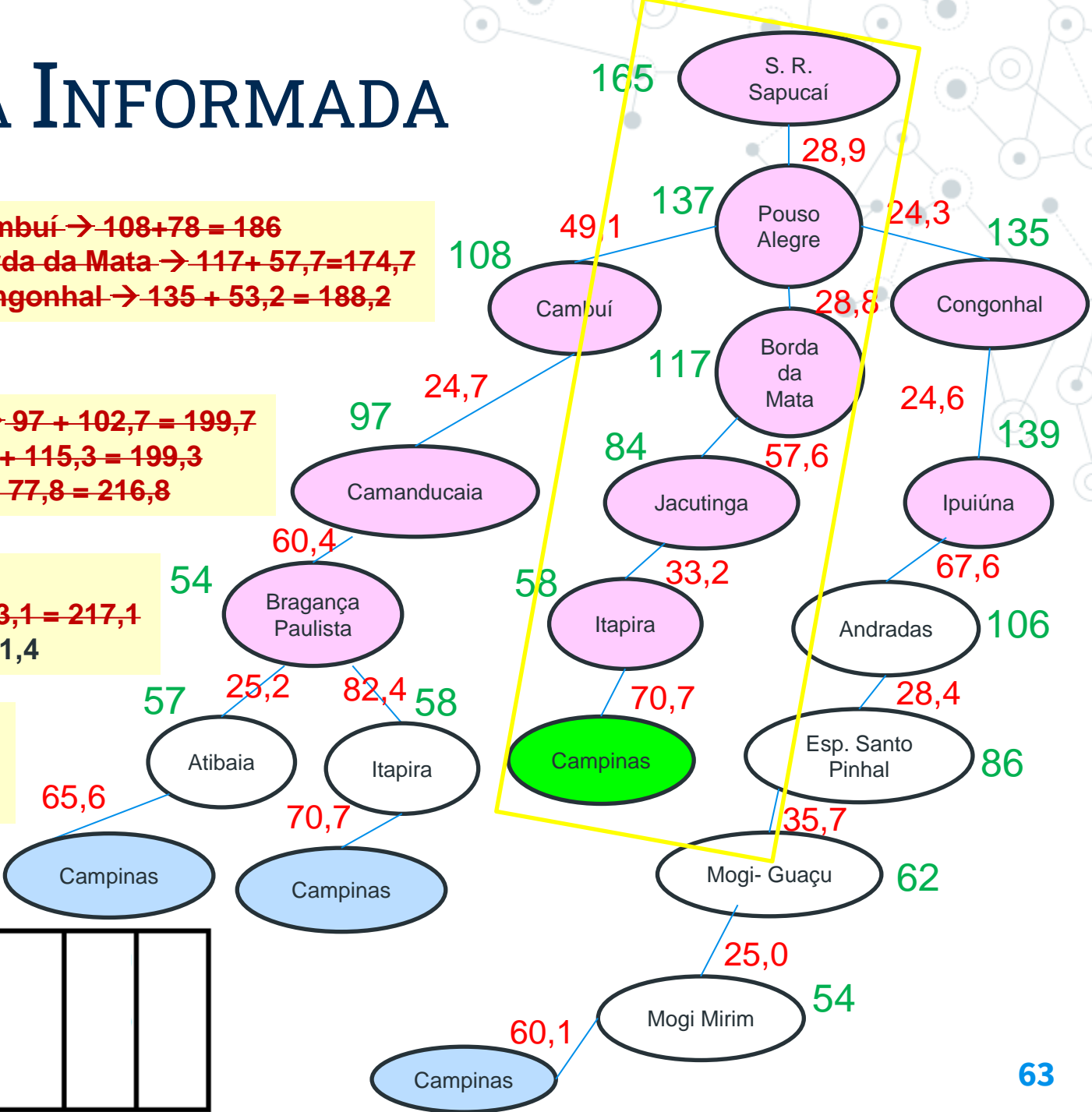
~~Camanducaia~~ → ~~97 + 102,7 = 199,7~~
~~Jacutinga~~ → ~~84 + 115,3 = 199,3~~
~~Ipuiúna~~ → ~~139 + 77,8 = 216,8~~

~~Itapira~~ → ~~58 + 148,5 = 206,5~~
~~Bragança Paulista~~ → ~~54 + 163,1 = 217,1~~
~~Andradas~~ → ~~106 + 145,4 = 251,4~~

~~Atibaia~~ → ~~57 + 188,3 = 245,3~~
~~Itapira~~ → ~~58 + 245,5 = 303,5~~
~~Campinas~~ → ~~0 + 219,2 = 219,2~~

Nós Visitados:

Sapucai	S. R.	Pouso Alegre	Borda da Mata	Cambuí	Congonhal	Jacutinga	Camanducaia	Itapira	Ipuiúna	Bragança Paulista	Campinas						
---------	-------	--------------	---------------	--------	-----------	-----------	-------------	---------	---------	-------------------	----------	--	--	--	--	--	--



RESUMO

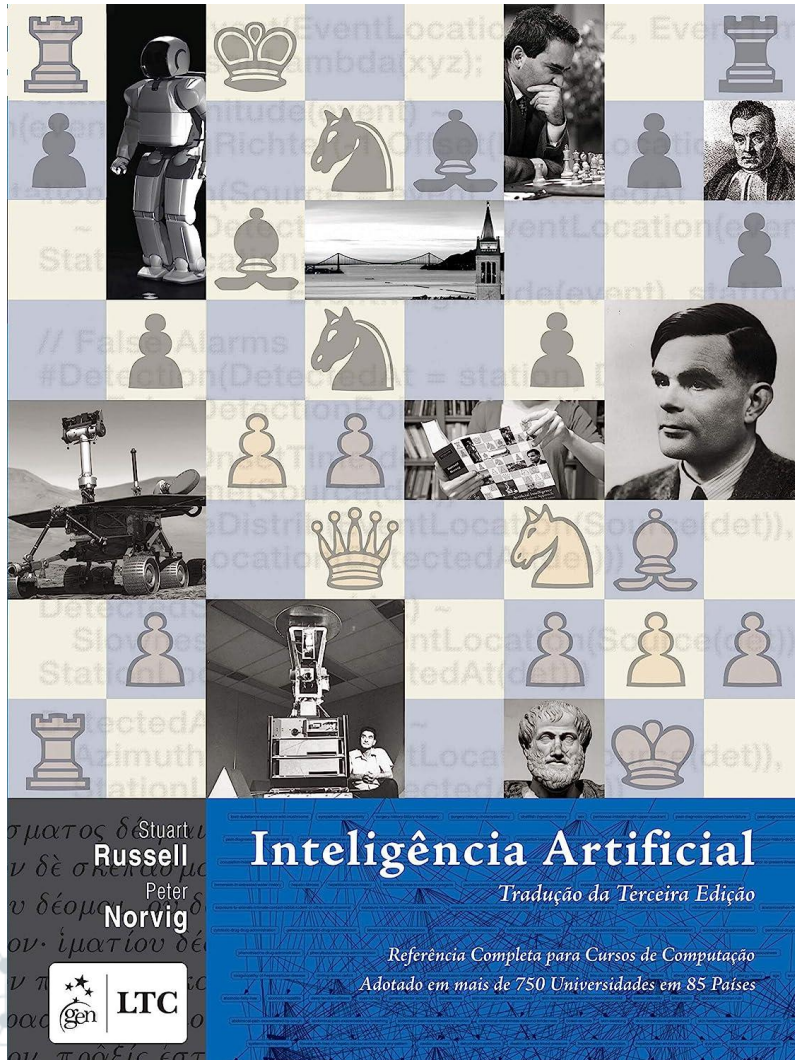
- Esta aula introduziu que um **agente** pode usar para **selecionar ações** em ambientes **determinísticos, observáveis, estáticos e completamente conhecidos**. Em tais casos, o agente pode construir **sequências de ações que alcançam seus objetivos**; esse processo é chamado de **busca**.
- Antes de um agente poder começar a procurar soluções, ele deve identificar um objetivo e formular um **problema bem definido**.
- Um problema consiste em cinco partes: **o estado inicial, um conjunto de ações, um modelo de transição descrevendo os resultados dessas ações, uma função teste de objetivo e uma função custo de caminho**. O **ambiente do problema** é representado por um **espaço de estados**. Um **caminho pelo espaço de estados** a partir do estado inicial até um estado objetivo é uma **solução**.
- Algoritmos de busca** tratam **estados** e **ações** como **atômicos**: sem considerar qualquer estrutura interna que possam ter.
- Um algoritmo genético de **BUSCA-EM-ÁRVORE** considera **todos os caminhos possíveis para encontrar uma solução**, enquanto um algoritmo de **BUSCA-EM-GRAFO** **evita a consideração de caminhos redundantes**.

RESUMO

- Os algoritmos de busca são analisados em termos de **completeza, otimização, complexidade de tempo e complexidade de espaço**. A complexidade depende de b , o *fator de ramificação no espaço de estados*, e de d , a *profundidade da solução mais rasa*.
- Métodos de busca não informados** têm acesso apenas à definição do problema. Os algoritmos básicos são:
 - A **Busca em Largura** seleciona para expansão os nós mais rasos; ela é completa, ótima para passos de custo unitário, mas tem complexidade de tempo exponencial.
 - A **Busca em Profundidade** expande o nó não expandido mais profundo. Ela não é completa nem ótima, mas tem complexidade espacial linear.
- Os **Métodos de busca informada** podem ter acesso a uma função heurística $h(n)$ que estima o custo de uma solução a partir de n .
 - O **algoritmo geral de busca de melhor escolha** seleciona um nó para a expansão de acordo com uma função de avaliação.
 - Uma **Busca A*** expande os nós com $f(n) = g(n) + h(n)$ mínimo. A* é completa e ótima, desde que $h(n)$ atenda a algumas restrições. A complexidade de espaço do A* ainda é proibitiva.

O desempenho de algoritmos de busca heurística depende da qualidade da função heurística.

REFERÊNCIAS



- RUSSELL, Stuart; NORVIG, Peter (Peter Norvig); SOUZA, Vandenberg Dantas De, Inteligência artificial. Rio de Janeiro, RJ: Editora Campus, 2004 - 2013, ISBN 978-85-352-1177-1 / 978-85-352-3701-6.

→ **Ler Capítulo 3**

The left side of the slide is dominated by a large, dark image of the Inatel building at night. The building's curved facade is covered in a grid of windows, many of which are illuminated from within, creating a warm, yellow glow. The Inatel logo is visible on the upper part of the building's facade.

Obrigada pela Atenção!

Dúvidas?

victoria.souto@Inatel.br