APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

ALGORITMOS DE BUSCA

AGENDA

- Componentes de um problema
- Exemplos
  - Mapa
  - Puzze 8
- Busca Gulosa
- Busca A* (“A estrela”)
- Estruturas de Dados

APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

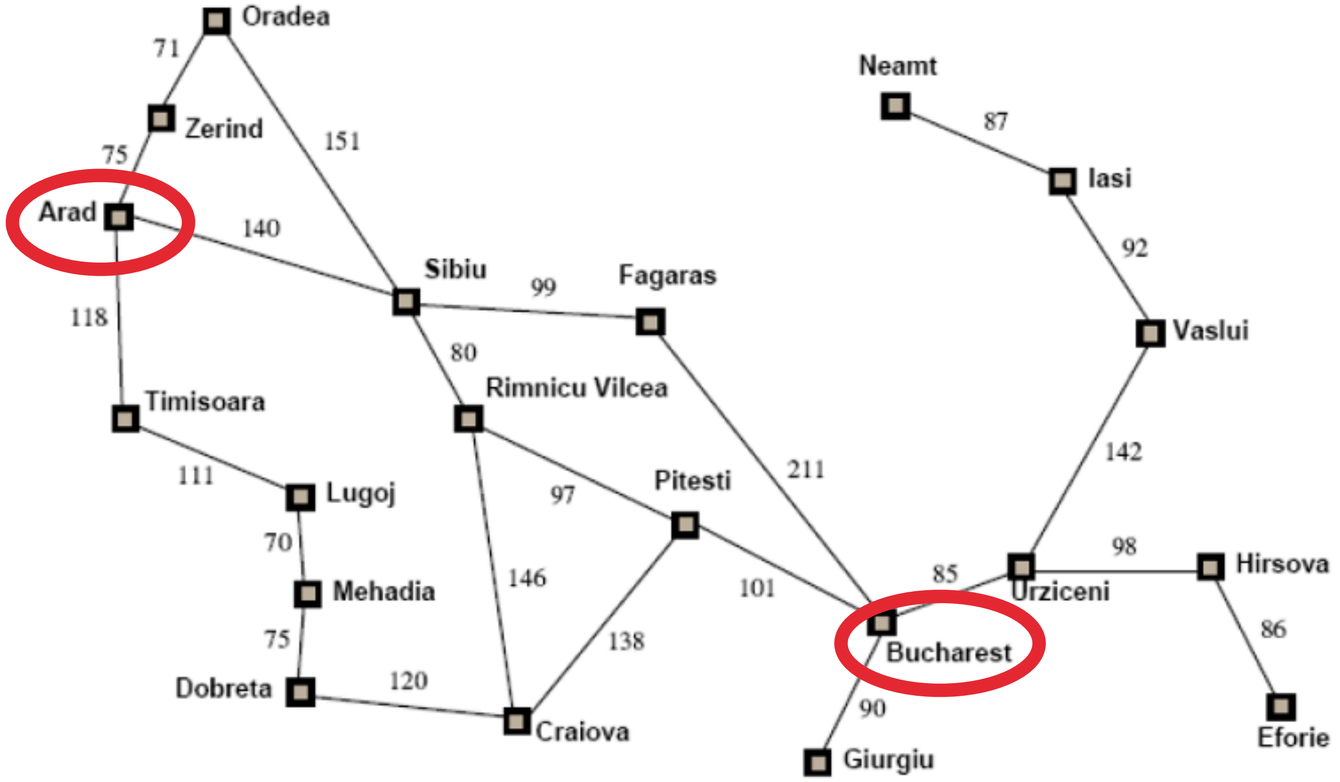
COMPONENTES DE UM PROBLEMA

COMPONENTES DE UM PROBLEMA

- Para solucionarmos problemas de busca precisamos de
  - Estado inicial –> Nosso ponto de partida
  - Estado final (objetivo)
  - Espaço de estados
  - Ações para passagem de um estado para outro
  - Solução (como chegar do estado inicial ao estado final)

EXEMPLO

Task: Find a path from Arad to Bucharest



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

6

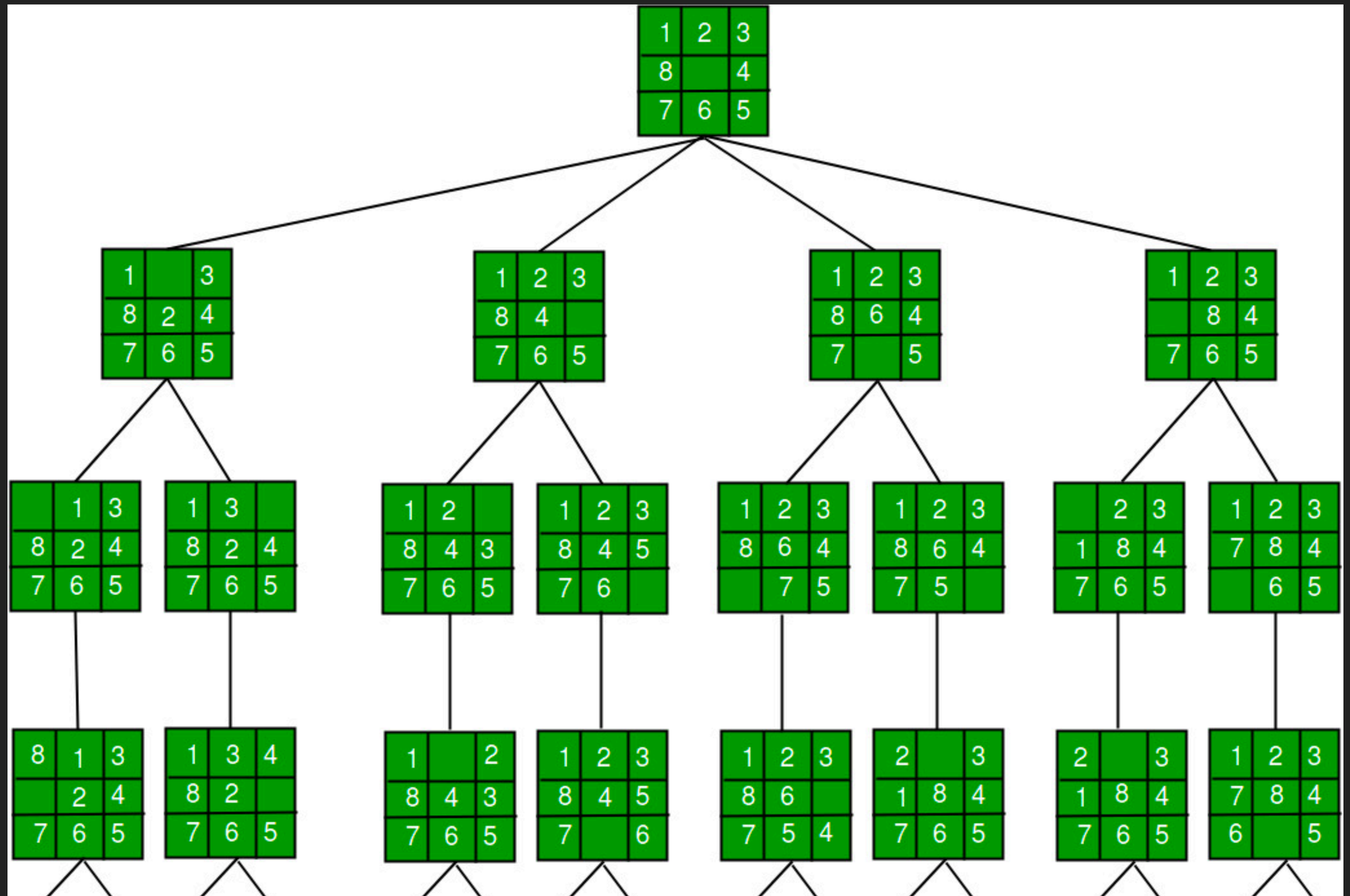
TAREFA: ENCONTRAR O MELHOR CAMINHO (PADRÃO) PARA IR DE "ARAD" ATÉ "BUCHAREST" (ROMÊNIA)

Fonte: Russell S. e Norvig P, Inteligência Artificial

EXEMPLO: ARAD -> BUCHAREST

- **Estado inicial:** Arad
- **Estado final (objetivo):** Bucharest
- **Espaço de estados:** Todos os possíveis caminhos entre as cidades
- **Ações para passagem de um estado para outro:** Os “custos" para passagem entre cada cidade
- **Solução:** Veremos (e implementaremos) o melhor caminho

EXEMPLO 2



TAREFA: ORDERNAR OS NÚMEROS DO PUZZLE

Fonte: Internet

EXEMPLO: 8 PUZZE

- **Estado inicial:** Tabuleiro desordenado
- **Estado final (objetivo):** Tabuleiro ordenado
- **Espaço de estados:** Todos os possíveis movimentos de peças
- **Ações para passagem de um estado para outro:** Movimento das peças. Cada movimento gera um novo espaço de estados
- **Solução:** Veremos as movimentações necessárias

EXEMPLO 3

TAREFA: GANHAR O JOGO DA VELHA

X	X	X
O	O	X
X	O	O

APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

HEURÍSTICAS

HEURÍSTICAS

- Imagine um jogo de xadrez –> Existem 10^{56} possibilidades de movimentos
  - Isso é superior ao número de elétrons no universo. Impossível de ser calculado, inclusive por um computador
- Para resolver este problema computacionalmente, é necessário delimitar o espaço de estados, uma opção: **Uso de heurísticas**
- É uma maneira de resolver um problema

HEURÍSTICAS

- Indicam as escolhas que o software deverá priorizar
- Através de cálculos matemáticos
- Existem técnicas de ajuda e descoberta para descobrir os melhores caminhos
- É como a brincadeira do “quente / frio”

HEURÍSTICA: EXEMPLO COM 8 PUZZE

1	8	2
	4	3
7	6	5

1	2	3
4	5	6
7	8	-

HEURÍSTICA: EXEMPLO COM 8 PUZZE

Estado inicial

1	8	2
	4	3
7	6	5

HEURÍSTICA: EXEMPLO COM 8 PUZZE

Possibilidade 1 em relação ao estado inicial (descer o 1)

	8	2
1	4	3
7	6	5

Custo dessa solução

Número de movimentos de cada peça: 1 + 1 + 1 + 1 + 2 + 2 + 0 + 2 = **10**

Peça: (1) (2) (3) (4) (5) (6) (7) (8)

HEURÍSTICA: EXEMPLO COM 8 PUZZE

Possibilidade 2 em relação ao estado inicial (4 para esquerda)

1	8	2
4		3
7	6	5

Custo dessa solução

Número de movimentos de cada peça: 0 + 1 + 1 + 0 + 2 + 2 + 0 + 2 = **8**

Peça: (1) (2) (3) (4) (5) (6) (7) (8)

HEURÍSTICA: EXEMPLO COM 8 PUZZE

Possibilidade 3 em relação ao estado inicial (subir o 7)

1	8	2
7	4	3
	6	5

Custo dessa solução

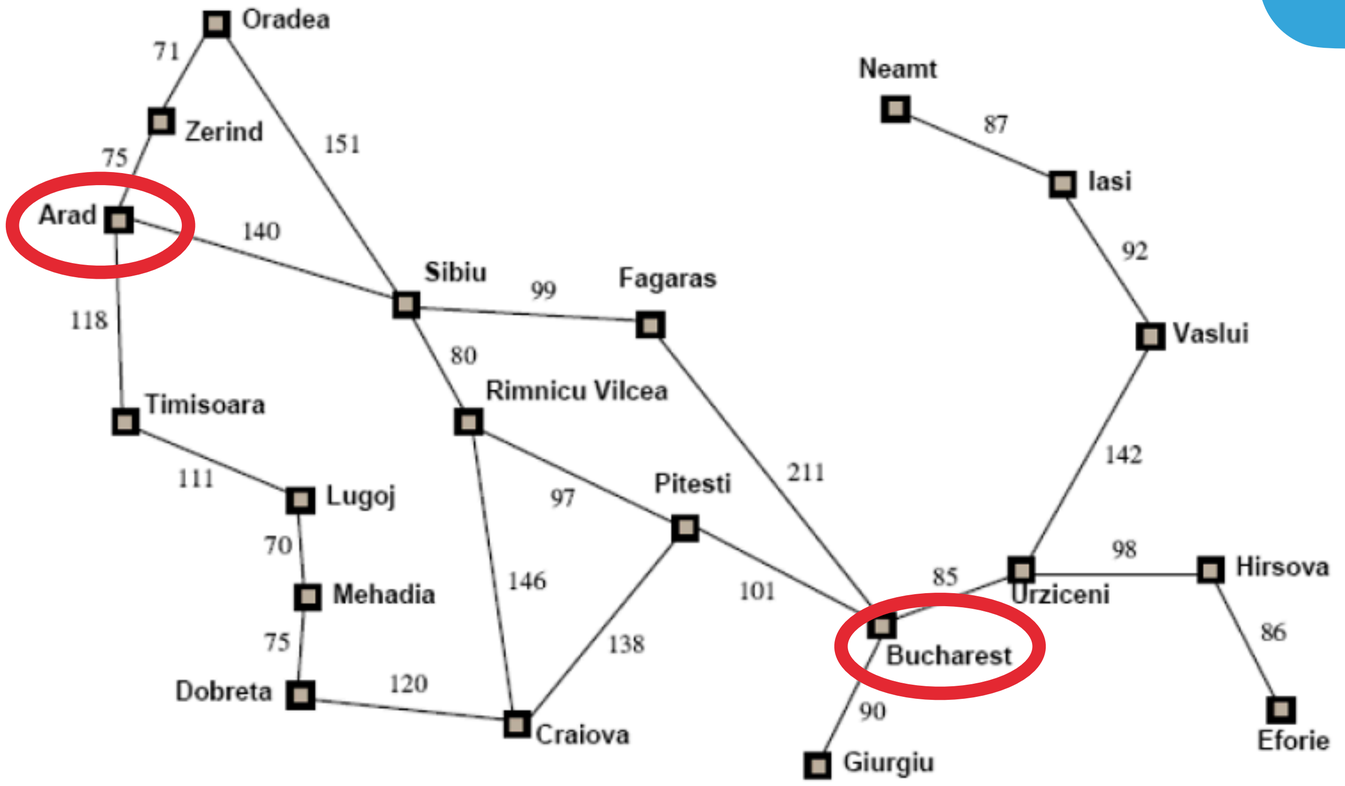
Número de movimentos de cada peça: 0 + 1 + 1 + 1 + 2 + 2 + 1 + 2 = **10**

Peças: (1) (2) (3) (4) (5) (6) (7) (8)

HEURÍSTICAS

- Neste pequeno exemplo, percebemos que apesar de existirem MUITAS soluções possíveis, podemos partir da segunda, ou seja, mover o 4 para esquerda
- Essa é a aplicação de uma heurística
- É apenas uma forma de resolver o problema (indica o quão perto você está na solução do problema
- Nem sempre é a melhor técnica, pois os estados seguintes podem gerar pesos maiores

EXEMPLO COM USO DE HEURÍSTICA



DISTÂNCIA EM LINHA RETA, ATÉ 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

6

TAREFA: ENCONTRAR O MELHOR CAMINHO (PADRÃO) PARA IR DE "ARAD" ATÉ "BUCHAREST" (ROMÊNIA)

Fonte: Russell S. e Norvig P, Inteligência Artificial

HEURÍSTICAS

- No exemplo das cidades, as distâncias tabela, entre cada cidade e Bucharest é uma heurística que pode ajudar o algoritmo a tomar a melhor decisão
- Lembre-se que não é a distância entre cidades, mas sim em linha reta (que não é o objetivo) mas pode ajudar a calcular a melhor solução
- É como o “quente frio” –> Ao analisar cada cidade você pode saber se está perto ou longe de Bucharest
- Portanto, para implementação, usaremos as distâncias em linha reta quanto as distâncias pelas estradas: Utilizaremos a **busca gulosa** e a **busca A***

HEURÍSTICAS

- Mas antes de implementarmos as buscas computacionalmente, temos que entender um pouco melhor sobre a **teoria dos vetores**
- Então vamos lá... teoria dos vetores e implementação...

APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

VETORES

VETORES

- Um vetor é uma estrutura de dados que armazena valores (normalmente do mesmo tipo), na memória e em uma única variável

Ex: V = [1, 5, 7, 3, 3, 7, 2]

- Este é um vetor de tamanho 7
- Portanto, suas posições endereçáveis são de 0 a 6 (n-1)
- Os valores internos são acessados pelo seu índica. Exemplo V[1] = 5 (neste caso)

VETORES ORDENADOS

- Um vetor ordenado significa que os valores da estrutura estão em ordem ascendente
- Ou seja, como menor valor no índice 0 e o maior na última posição.
- Cada índice sempre armazena um valor maior do que seu índice anterior
- v[0] <= v[1] <= ...<= v[N-1]
- Ex: V = [1, 3, 5, 6, 8, 9, 34]
- Vetores ordenados agilizam o tempo de pesquisa linear e binária

VETORES ORDENADOS – OPERAÇÕES – INSERÇÃO

- Em um vetor não ordenado a inserção é pouco custosa, pois pode ser realizada em qualquer índice da estrutura (normalmente ao final)
- Porém, em vetores ordenados, a inserção precisa manter sua ordenação, o gera um certo esforço computacional

VETORES ORDENADOS – OPERAÇÕES – INSERÇÃO

- Exemplo: V = [1, 3, 4, 5, 8, 9]; Inserir o número 7
- É preciso comparar o valor de cada índice e o próximo com o “7”, para saber em qual posição ele será incluído.
- É preciso alterar o índice de todos os elementos seguintes a identificação do local da inserção (i = i +1)
- Neste caso entre o 7 deverá ser inserido no índice 4 (onde está o número 8)
- Assim...

VETORES ORDENADOS – OPERAÇÕES – INSERÇÃO

7

Passo	[0]	[1]	[2]	[3]	[4]	[5]	[6]
1	1	3	4	5	8	9	
2	1	3	4	5		8	9
3	1	3	4	5	7	8	9

VETORES ORDENADOS – OPERAÇÕES – INSERÇÃO

- A pesquisa termina quanto o primeiro item maior que o valor inserido é atingido
- Os piores casos são de inserção no início e no fim
  - No início é fácil inserir, mas deve-se alterar o índice de TODOS os elementos do vetor
  - Ao final é simples de inserir, mas a pesquisa acaba percorrendo o vetor inteiro

VETORES ORDENADOS – OPERAÇÕES – BUSCA

- Existem, essencialmente, dois algoritmos de busca para vetores ordenados
  - **Busca linear**: Percorre índice por índice até encontrar o valor ou um valor maior
  - **Busca binária**: divide-se repetidamente pela metade a porção da lista que **deve** conter o item, até se reduzir as localizações a apenas uma possível
- Vamos analisar a diferença de uma busca linear de uma busca binária

https://www.cs.usfca.edu/~galles/visualization/Search.html

VETORES ORDENADOS – OPERAÇÕES – EXCLUSÃO

- Assim como a inserção, a exclusão deve realizar uma busca no vetor
- Assim que encontrar (se encontrar) o elemento excluido, ele deve ser substituído por todos os seguintes (se houver)
- A operação gera um grande esforço computacional de busca se
  - O item excluído não existir (devido ao esforço da busca)
  - O item excluído estiver no início (pois todos os elementos deverão ser movidos)

VETORES ORDENADOS – OPERAÇÕES – EXCLUSÃO

7

Passo	[0]	[1]	[2]	[3]	[4]	[5]	[6]
1	1	3	4	5	7	8	9
2	1	3	4	5		8	9
3	1	3	4	5	8		9
4	1	3	4	5	8	9	

VETORES ORDENADOS - IMPLEMENTAÇÃO

- Para entendermos um pouco melhor o processo todo, vamos implementar, em Python, um vetor ordenado

- Vamos lá... mãos à obra!

Arquivo: Aula - Vetores Ordenados

APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

BUSCA GULOSA

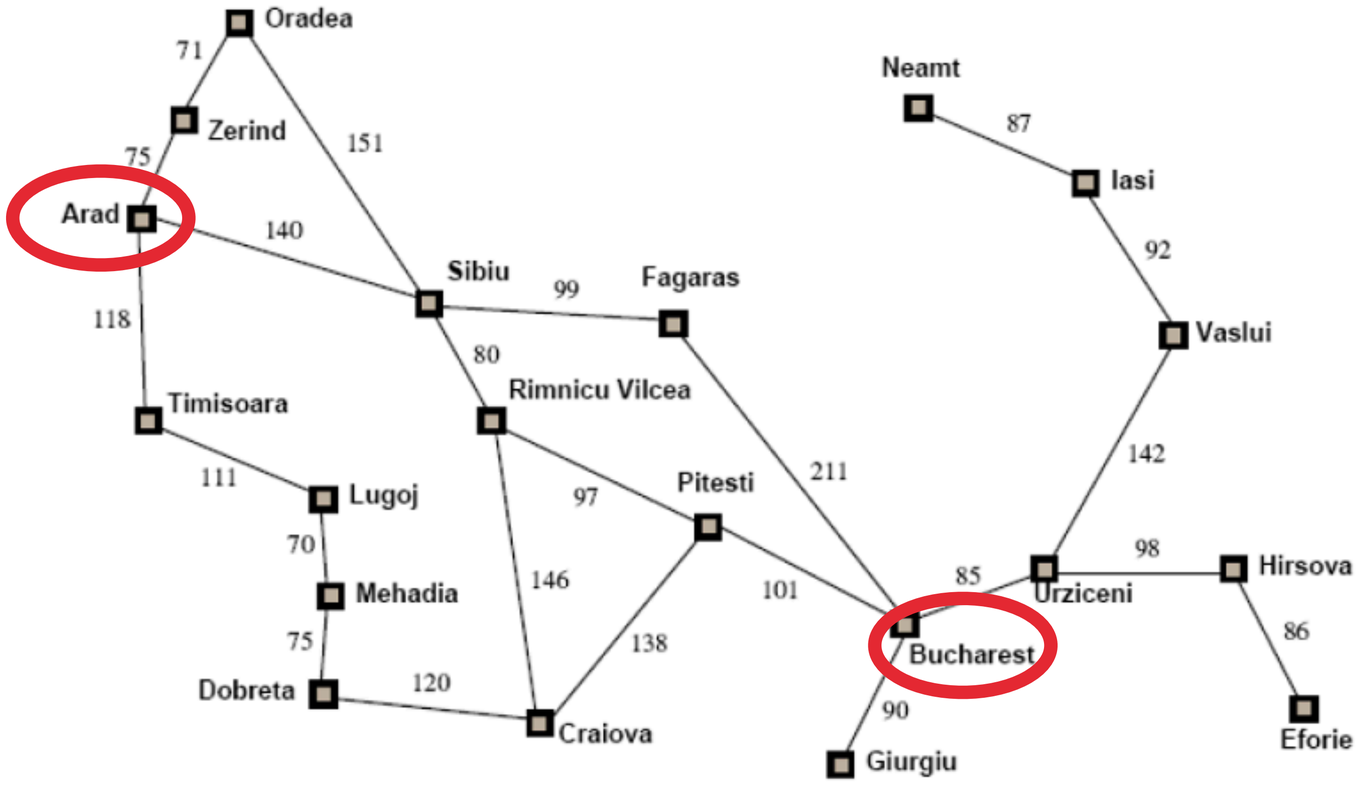
BUSCA GULOSA

- A Busca Gulosa utiliza o conceito de heurística
- Onde sabe-se algo sobre o problema a ser resolvido
- Para entender este algoritmo, vamos utilizar um estudo de caso, da viagem de Arad à Bucharest.
- Estudo de caso clássico de Stuart Russel e Peter Norvig - Livro “Inteligência Artificial”

BUSCA GULOSA – ESTUDO DE CASO

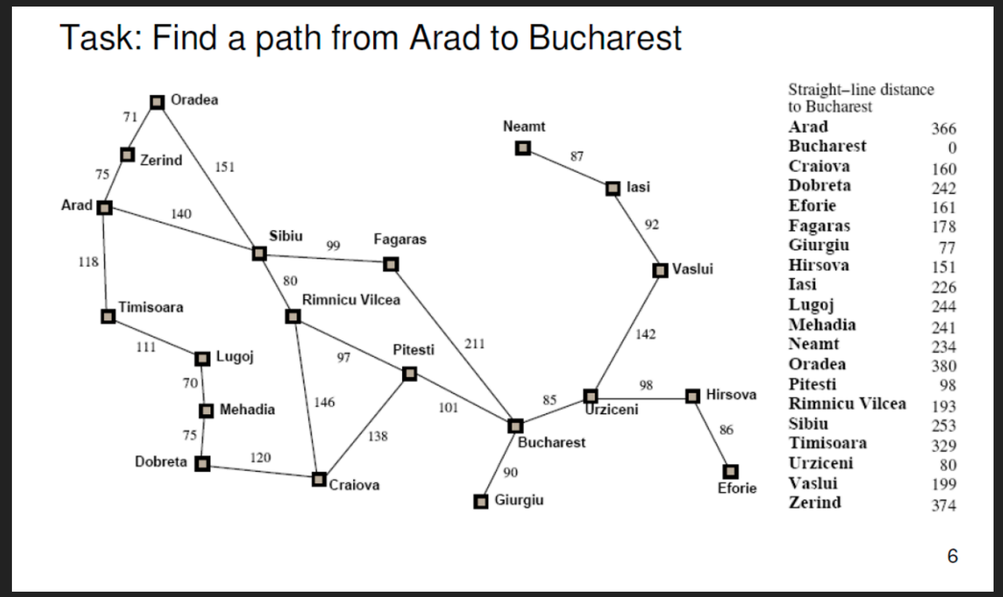
UTILIZAREMOS ESSAS INFORMAÇÕES

Task: Find a path from Arad to Bucharest



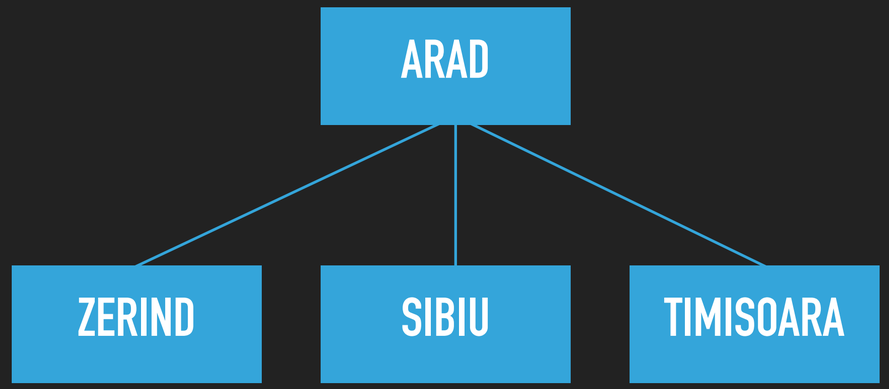
Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

BUSCA GULOSA – ESTUDO DE CASO

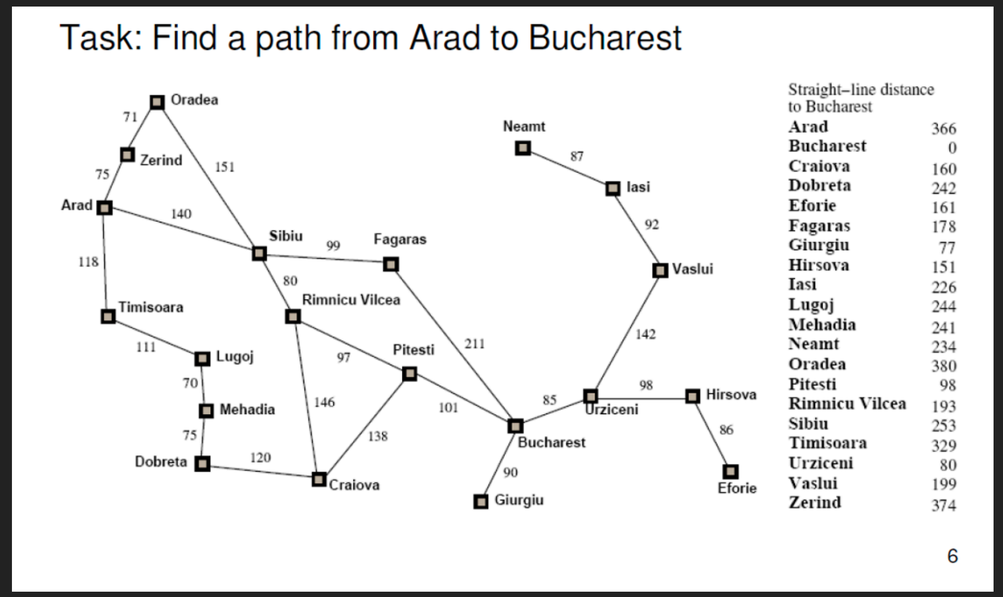


1° passo

Encontrar os adjacentes

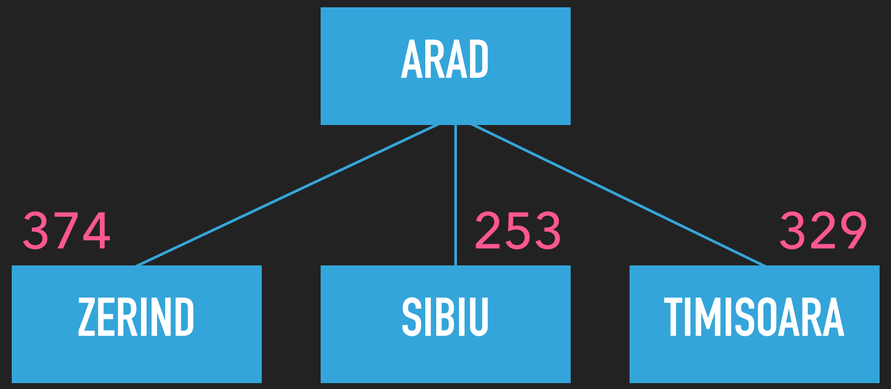


BUSCA GULOSA – ESTUDO DE CASO

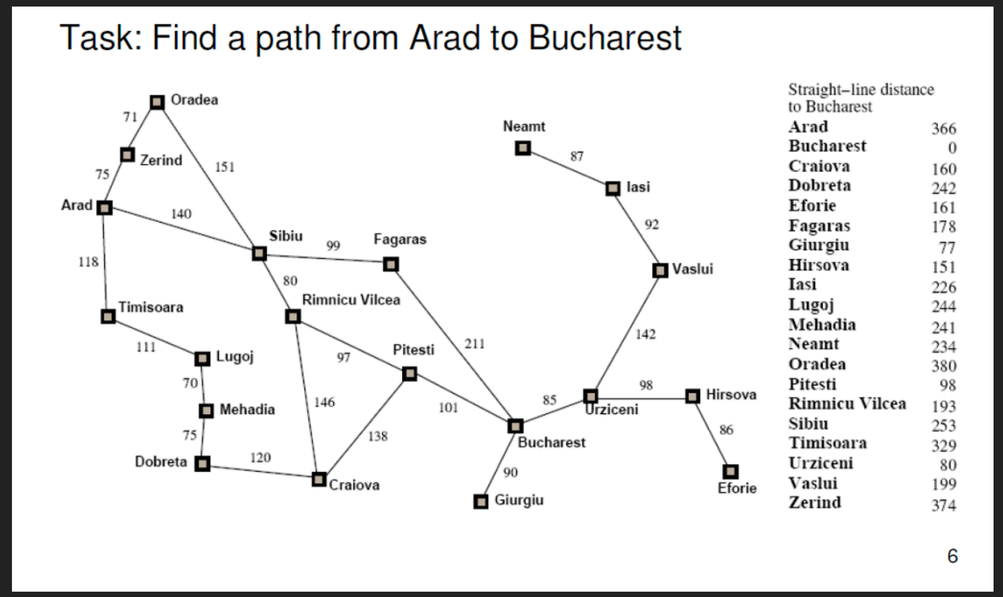


2° passo

Encontrar as distâncias em linha reta de cada adjacente ao objetivo



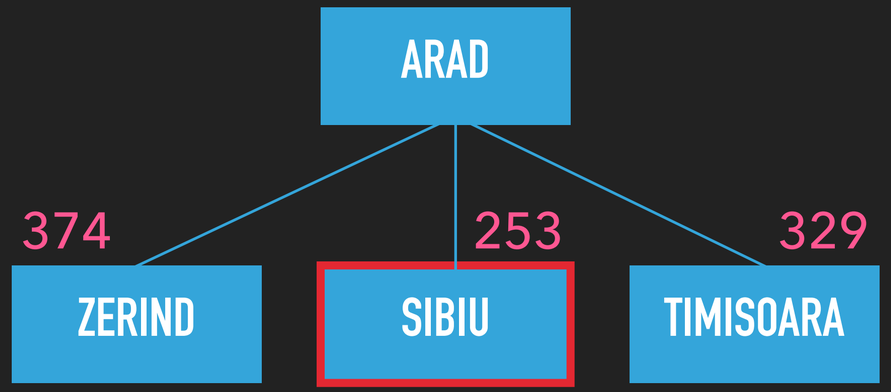
BUSCA GULOSA – ESTUDO DE CASO



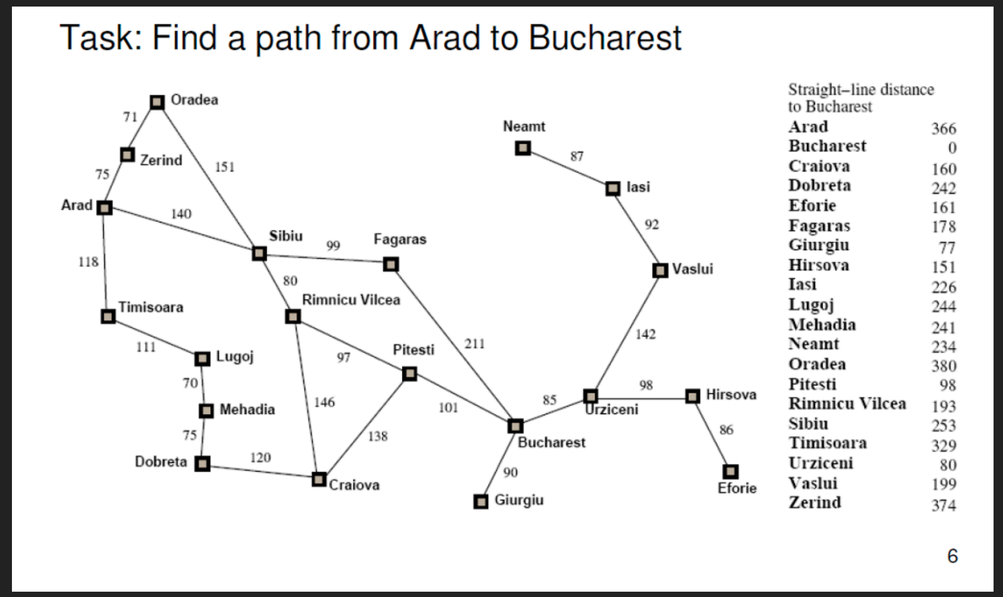
3° passo

Descobrir qual a menor distância.

Continuar a partir desta

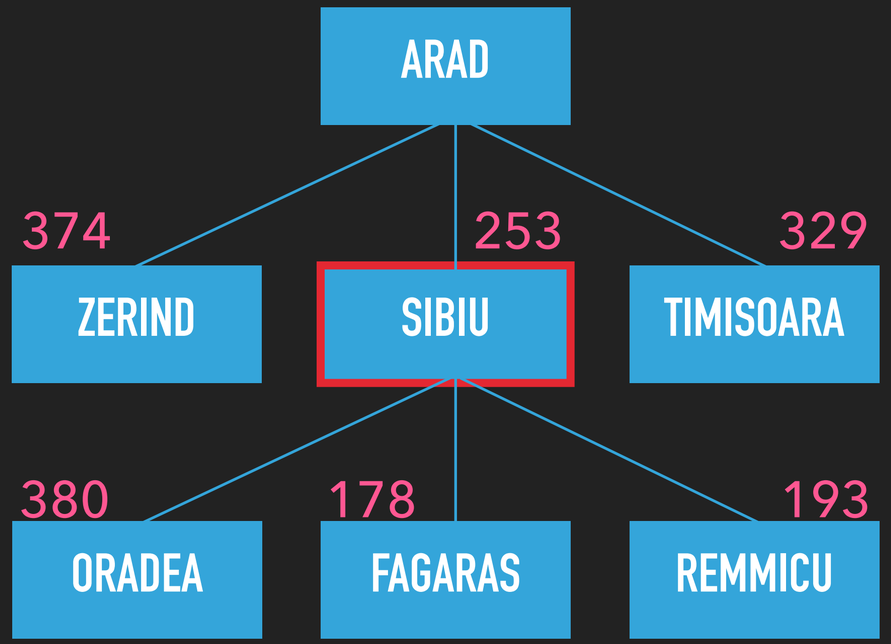


BUSCA GULOSA – ESTUDO DE CASO

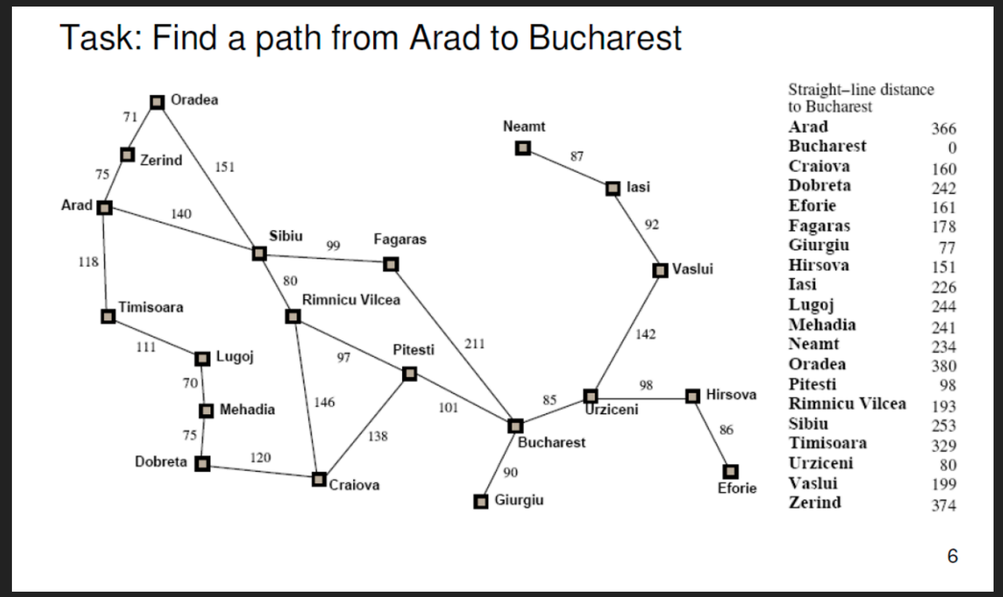


4° passo

Repetir o processo...

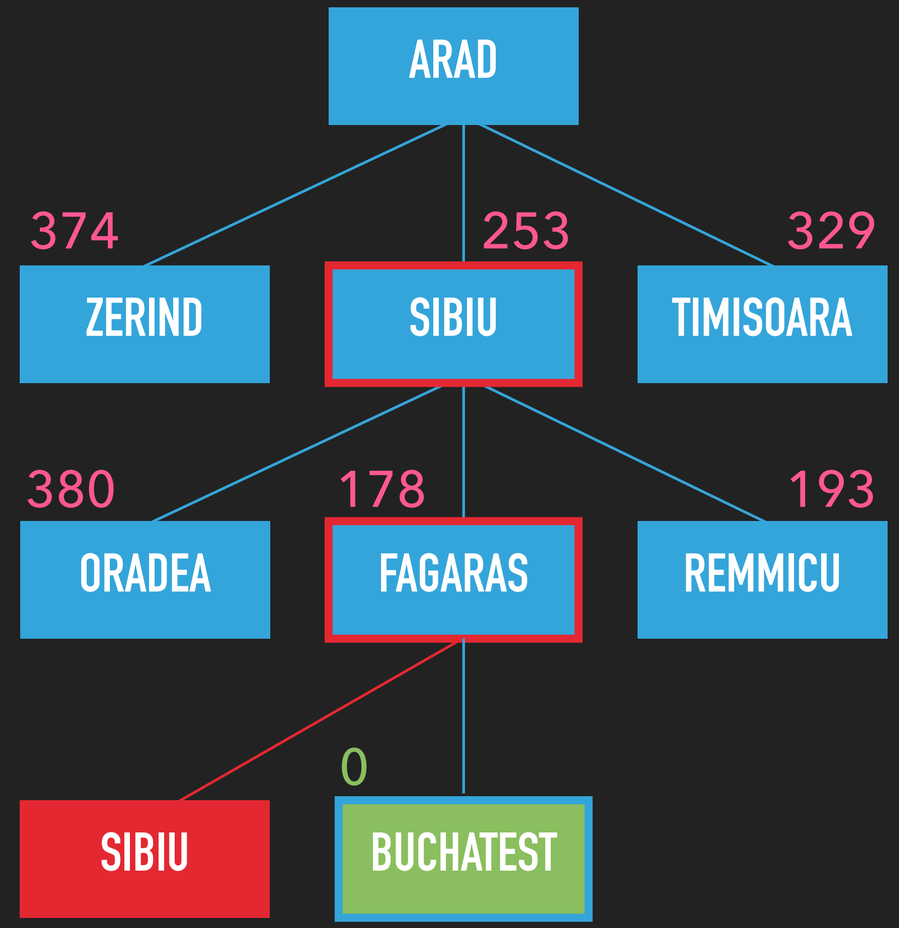


BUSCA GULOSA – ESTUDO DE CASO



5° passo

Continuar



Não verificamos nós já visitados

BUSCA GULOSA – ALGUMAS CONSIDERAÇÕES

- Essa solução é muito boa para mapas quando existem informações de distância em linha reta
- Em distâncias espaciais e considerando a curvatura da terra (sim, a terra é redonda), deve-se utiliza mais dados, como altitude, periculosidade, gastos de combustível etc.
- Por isso é tão importante termos em mãos os vetores ordenados com os dados utilizados para a decisão

BUSCA GULOSA – IMPLEMENTAÇÃO

- Vamos implementar agora
  - O grafo
  - A busca Gulosa

- Mãos a obra!

Arquivo: Aula - Grafo com Busca Gulosa

APLICAÇÕES EM INTELIGÊNCIA ARTIFICIAL

BUSCA A*

BUSCA A*

- Mais precisa do que a busca gulosa, pois considera mais de um fator para avançar
- Podem ser considerados fatores diversos na hora de realizar a busca, usando mais de um critério para a decisão
- MUITO usado em sistemas GPS reais e em mapas de jogos e para “inimigos" lhe perseguirem
-

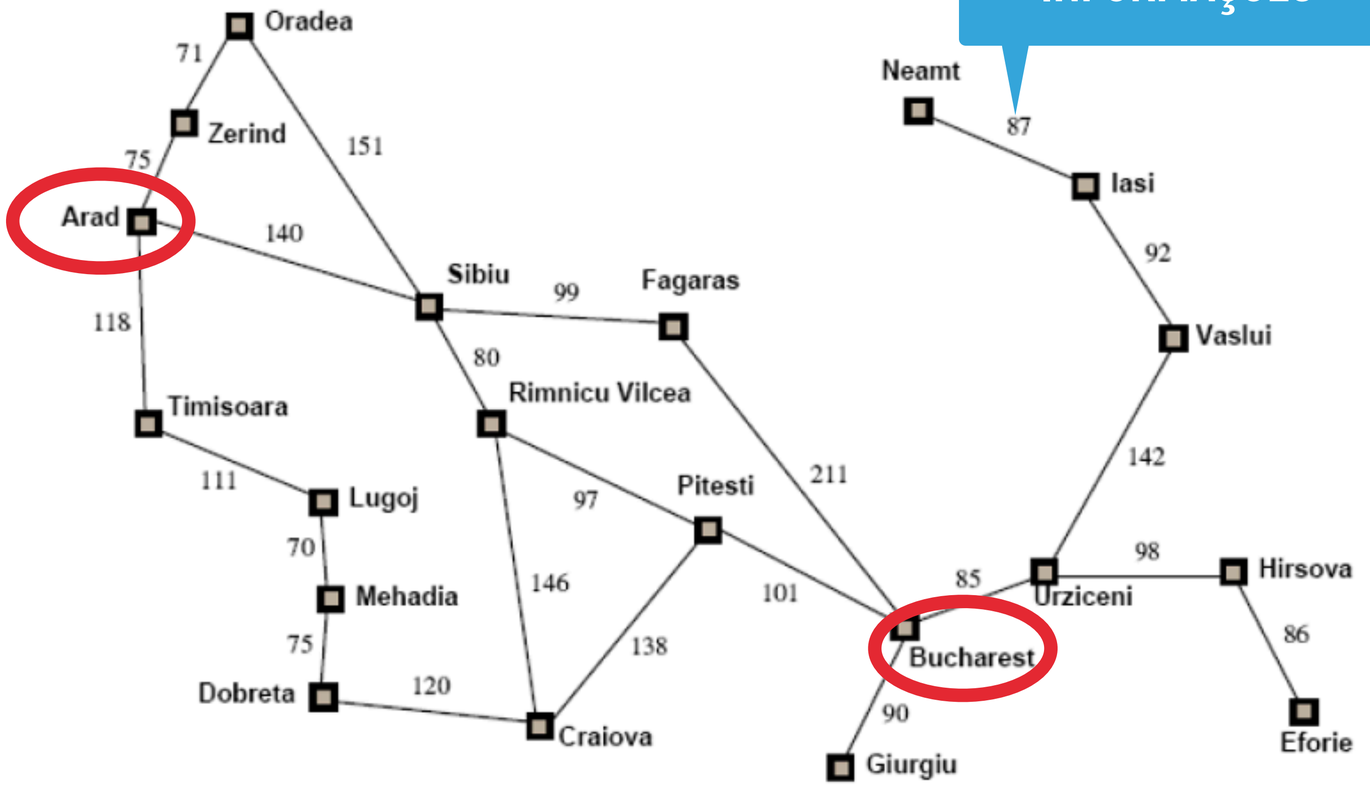
BUSCA A*

- No caso do exemplo da busca da rota entre “Arad" e “Bucharest”, vamos utilizar as duas informações:
  - A distância em linha reta E
  - A distância entre as cidades

BUSCA A* – ESTUDO DE CASO

UTILIZAREMOS ESSAS INFORMAÇÕES

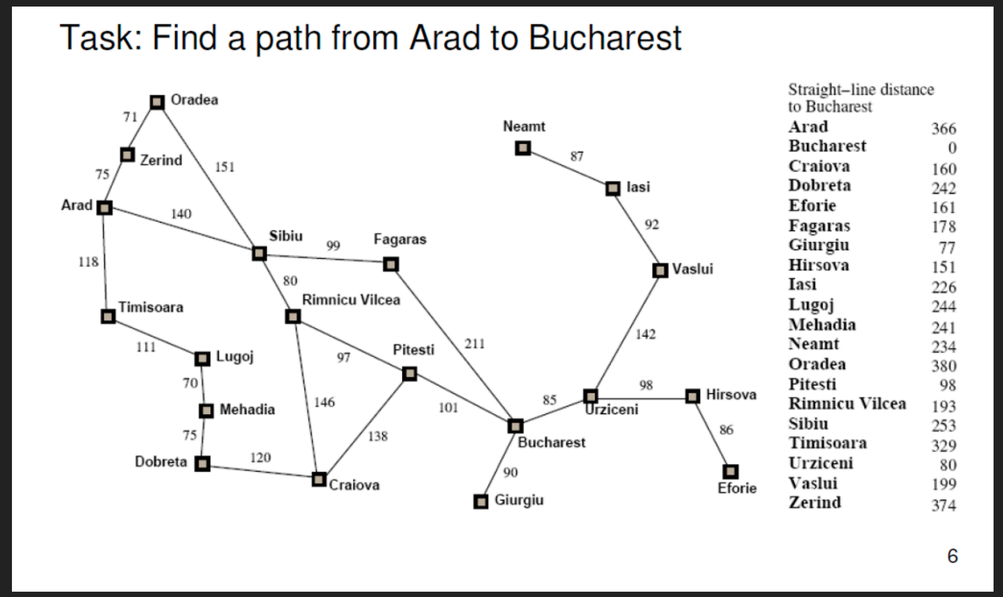
Task: Find a path from Arad to B



UTILIZAREMOS ESSAS INFORMAÇÕES

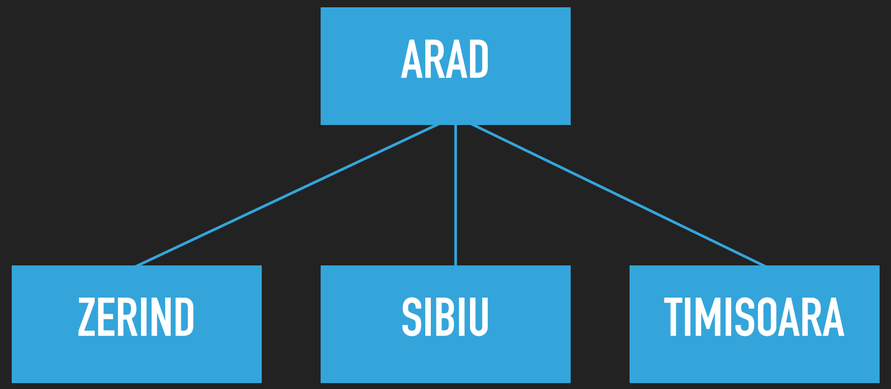
Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

BUSCA A* – ESTUDO DE CASO

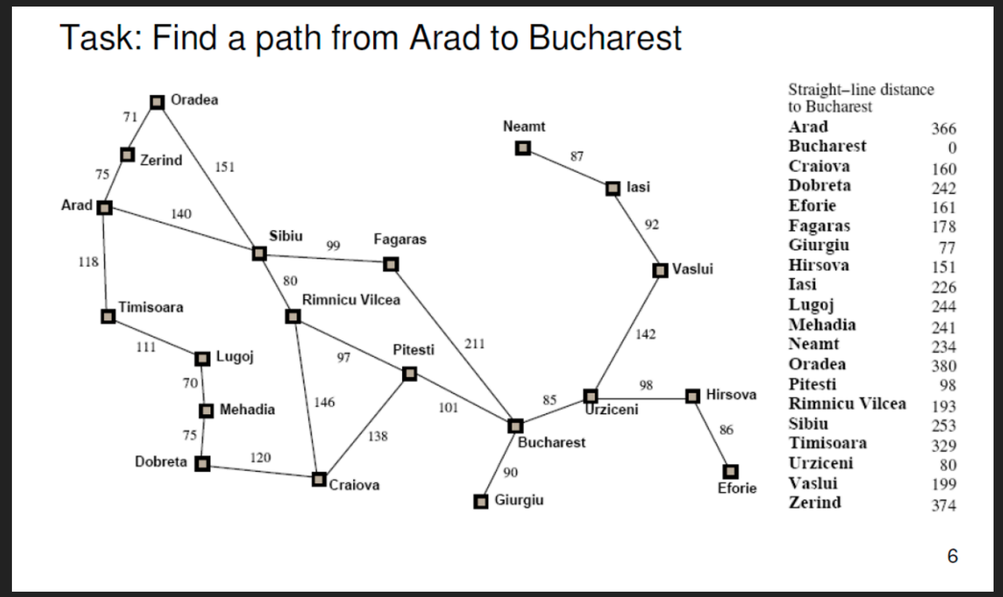


1° passo

Encontrar os adjacentes

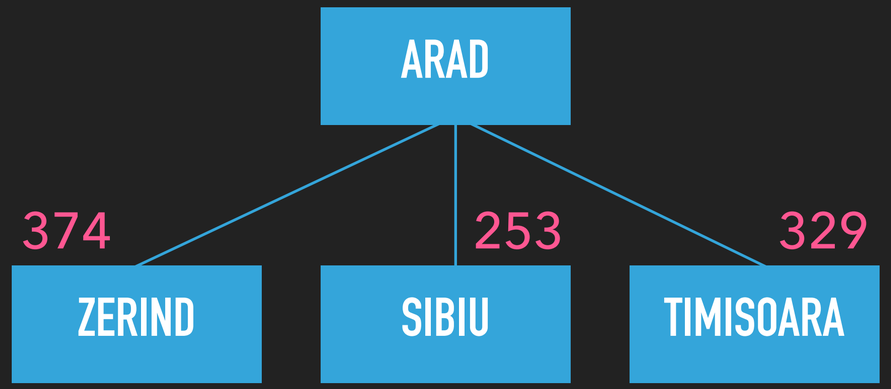


BUSCA A* – ESTUDO DE CASO

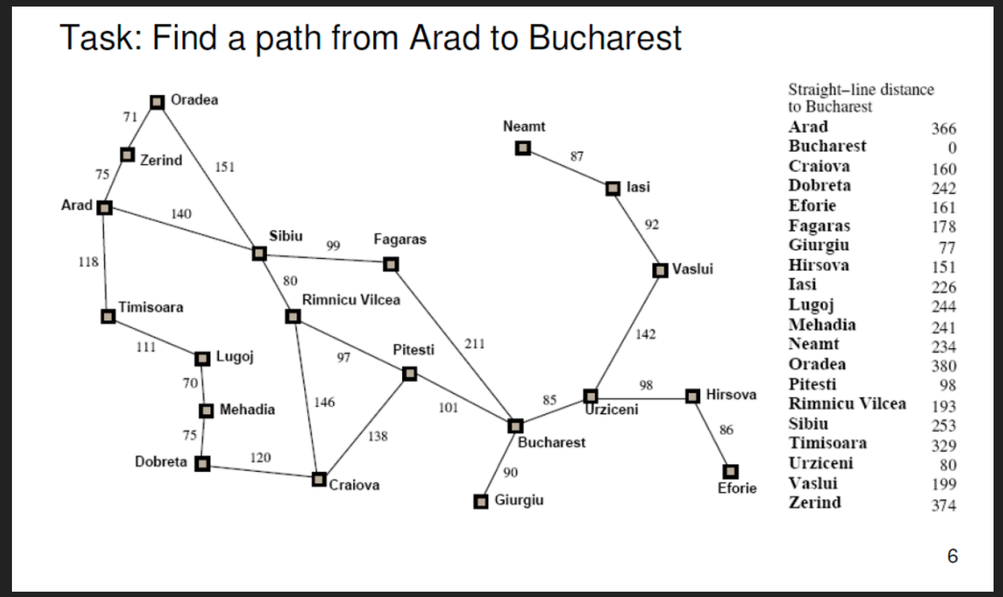


2° passo

Encontrar as distâncias em linha reta de cada adjacente ao objetivo

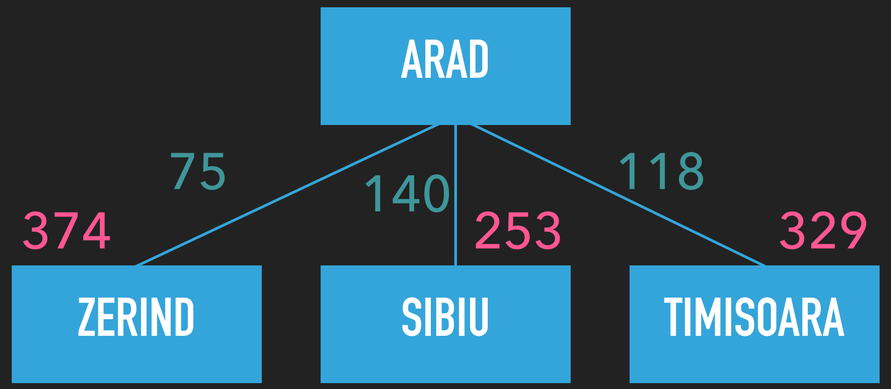


BUSCA A* – ESTUDO DE CASO

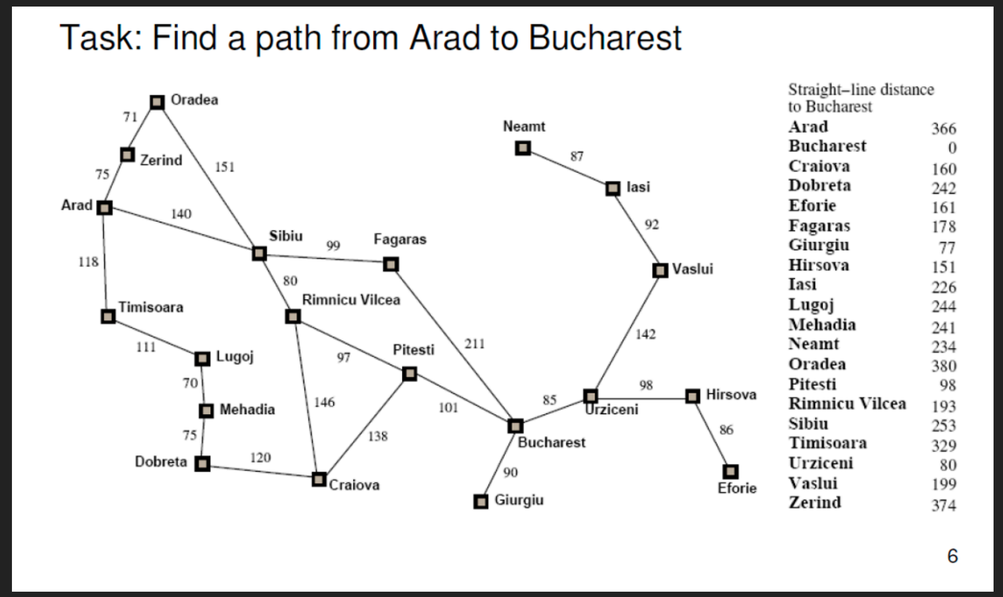


3° passo

Encontrar as distâncias também pela estrada

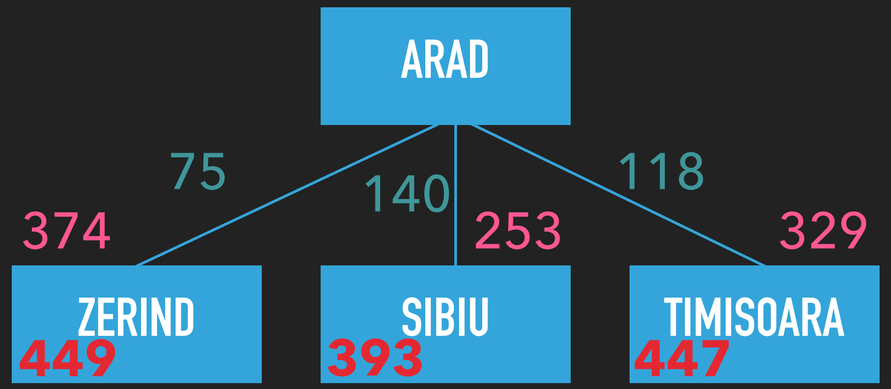


BUSCA A* – ESTUDO DE CASO

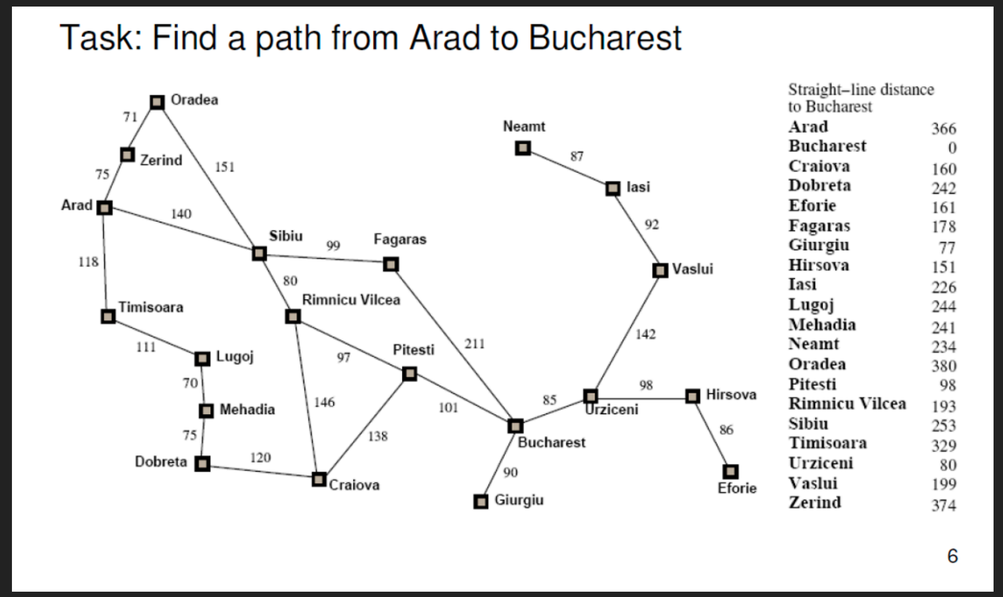


4° passo

Somar os valores e verificar qual o menor

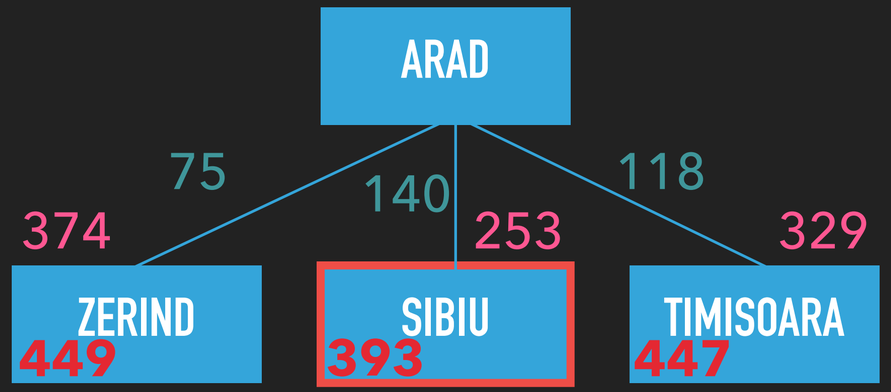


BUSCA A* – ESTUDO DE CASO

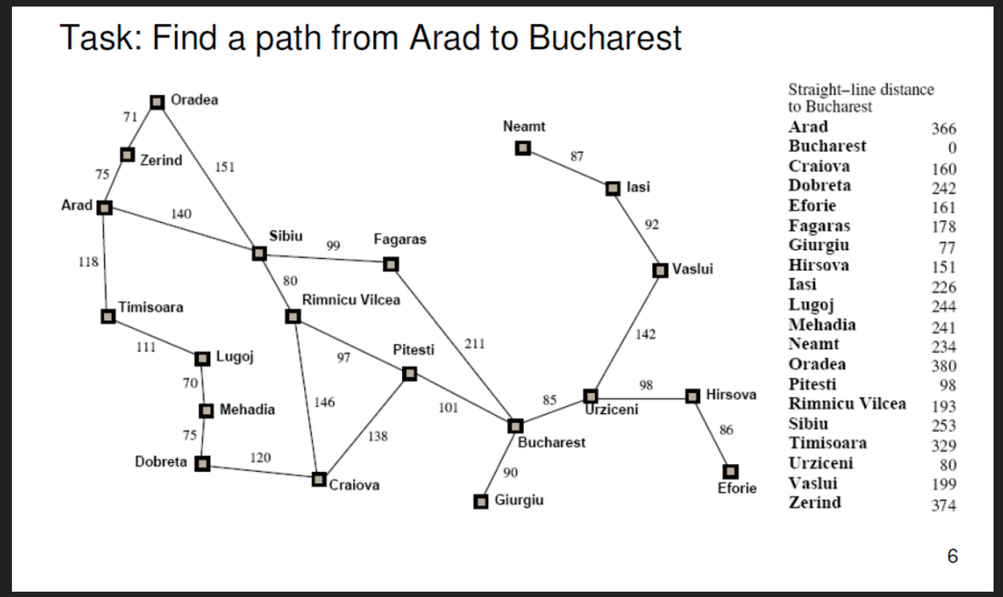


5° passo

Ir até o menor resultado. Neste caso, Sibiu

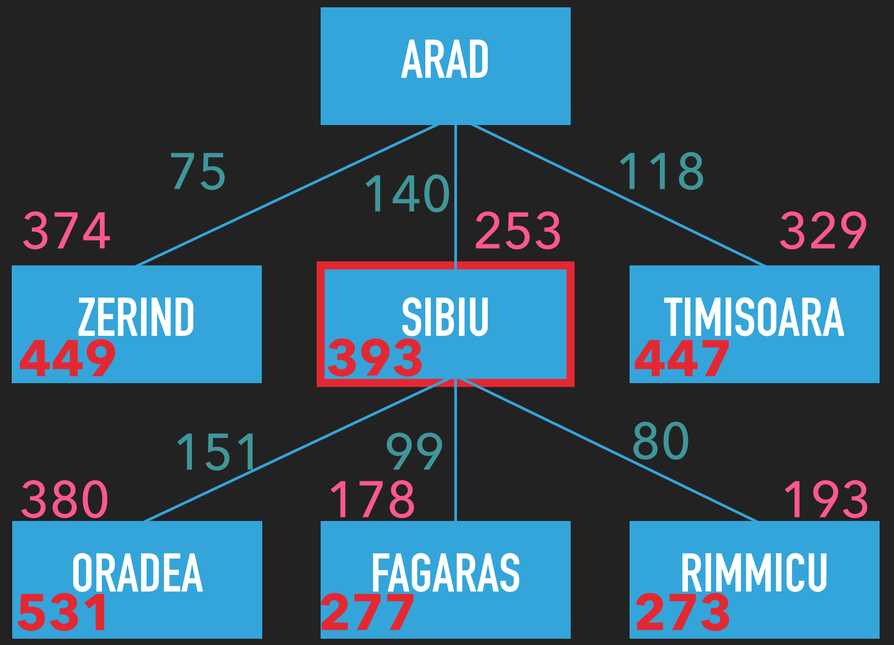


BUSCA A* – ESTUDO DE CASO

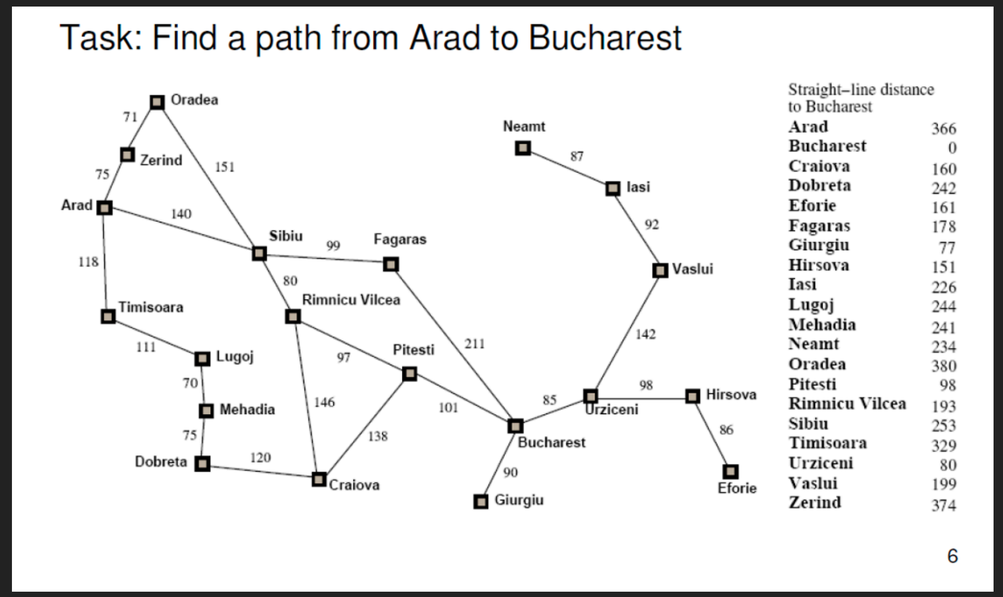


6° passo

Repetir o processo...

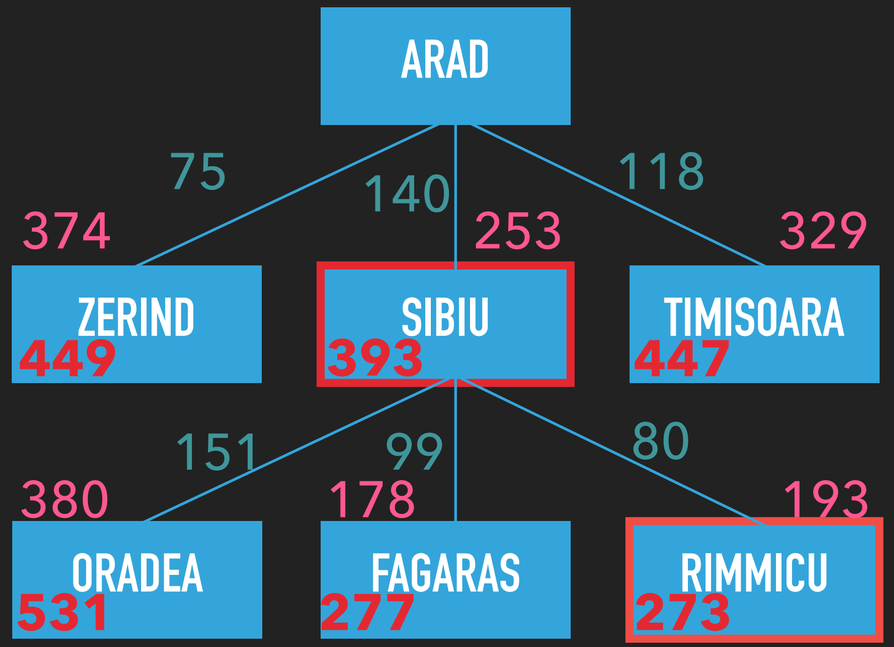


BUSCA A* – ESTUDO DE CASO

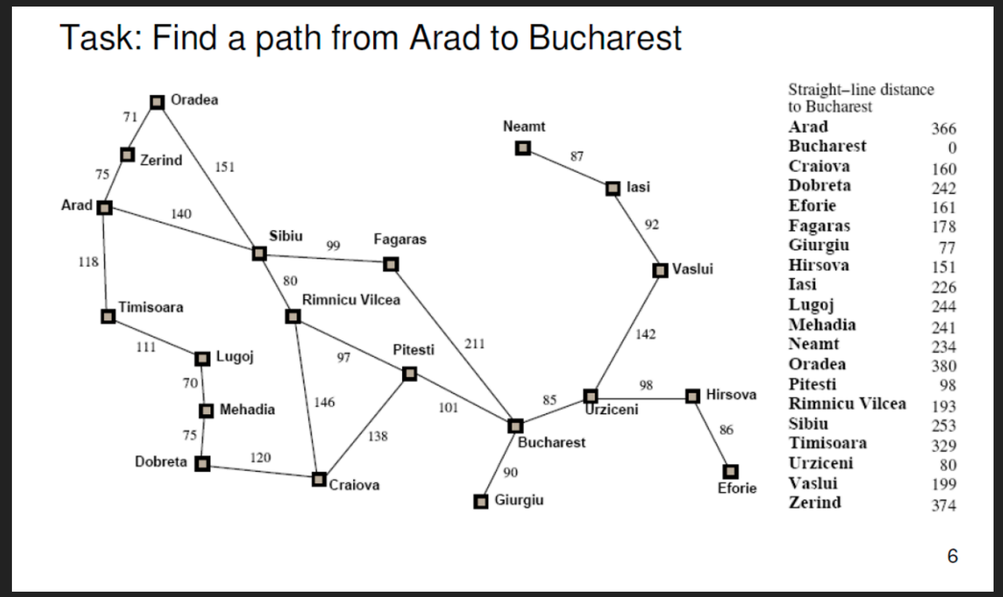


...

Repetir o processo...

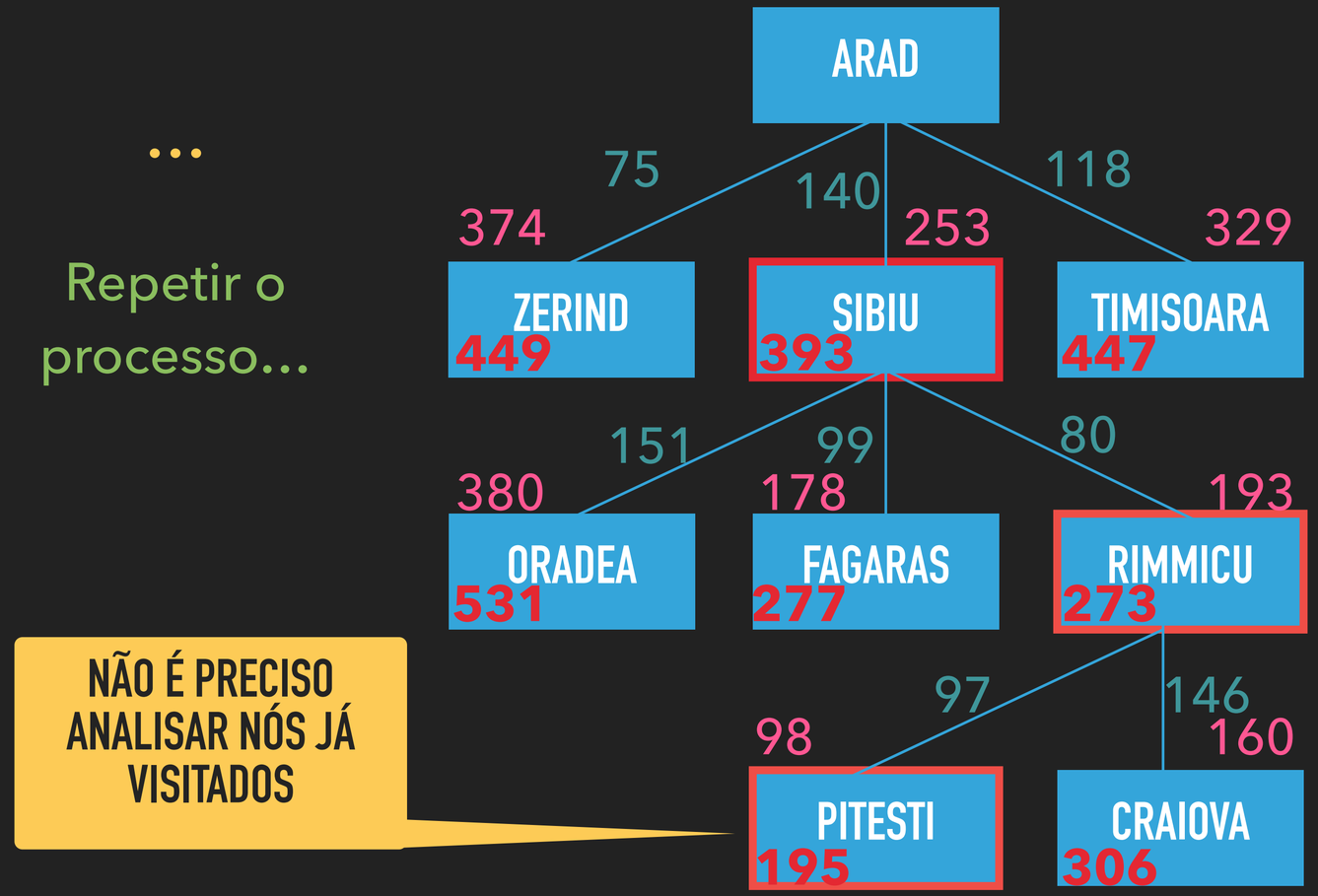


BUSCA A* – ESTUDO DE CASO



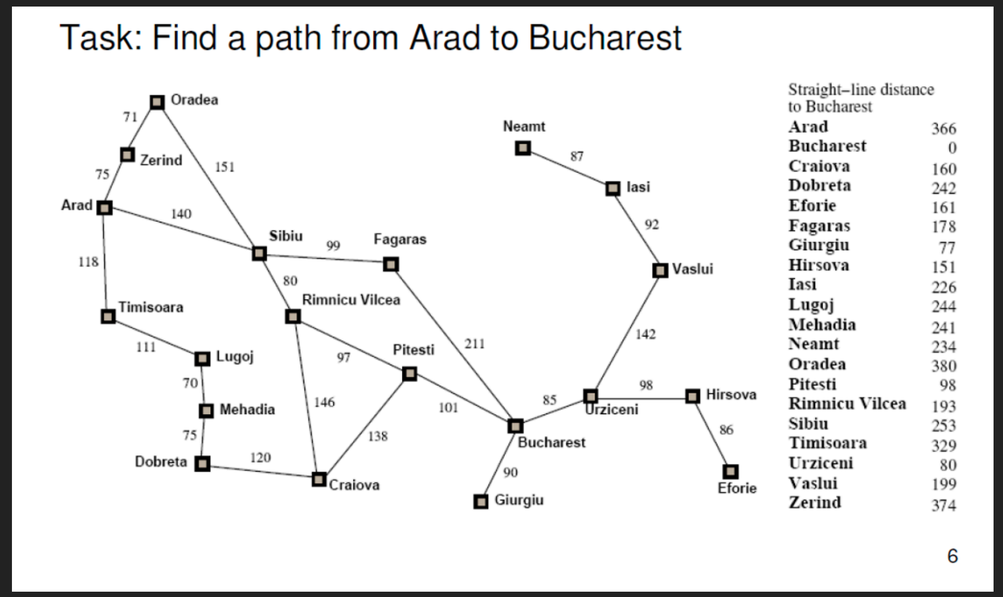
...

Repetir o processo...



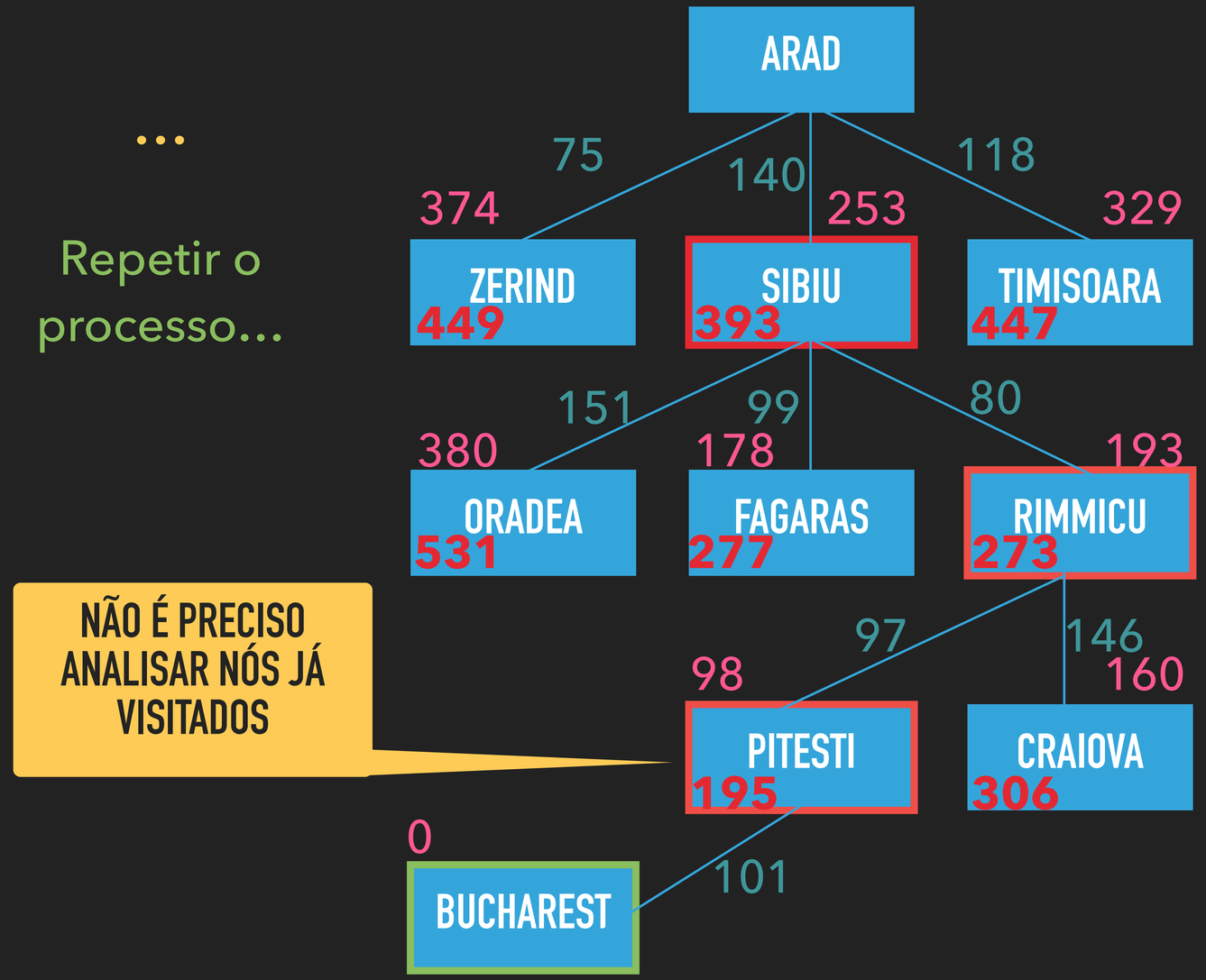
NÃO É PRECISO ANALISAR NÓS JÁ VISITADOS

BUSCA A* – ESTUDO DE CASO



...

Repetir o processo...



BUSCA A* – ALGUMAS CONSIDERAÇÕES

- Na busca gulosa o algoritmo gera um caminho de 450 Km
- Na busca A* o algoritmo gera um caminho de 418 Km
- Uma diferença de 32 Km
- Portanto a busca A* é mais precisa (embora computacionalmente mais lenta), pois considera-se mais fatores
- Podem ser considerados mais fatores ainda, como Tempo gasto para atravessar cada cidade, custos com pedágios, periculosidade etc.

BUSCA A* – IMPLEMENTAÇÃO

- Vamos implementar agora
  - O grafo (já o fizemos, vamos apenas atualizá-lo)
  - A busca A*
  - Mãos a obra!

Arquivo: Aula - Grafo com Busca A Estrela