



Agent de résolution de problèmes Stratégies d'exploration informée

Aziz khamjane

Plan

2

- Introduction
- Meilleur d'abord (BFS - *Best-first*)
- Meilleur d'abord gloutonne (*Greedy best-first*)
- Exemples
- A* (*A-Star*)
- Exemples
- Propriétés de A*
- Algorithmes heuristiques à mémoire limitée
- Génération des heuristiques

Introduction

- Une heuristique est utilisée pour guider la recherche :
 - les heuristiques exploitent les connaissances du domaine d'application.
- Les heuristiques sont à la base de beaucoup de travaux en IA :
 - Recherche de meilleures heuristiques
 - Apprentissage automatique d'heuristiques

Recherche: meilleur d'abord (best first search)

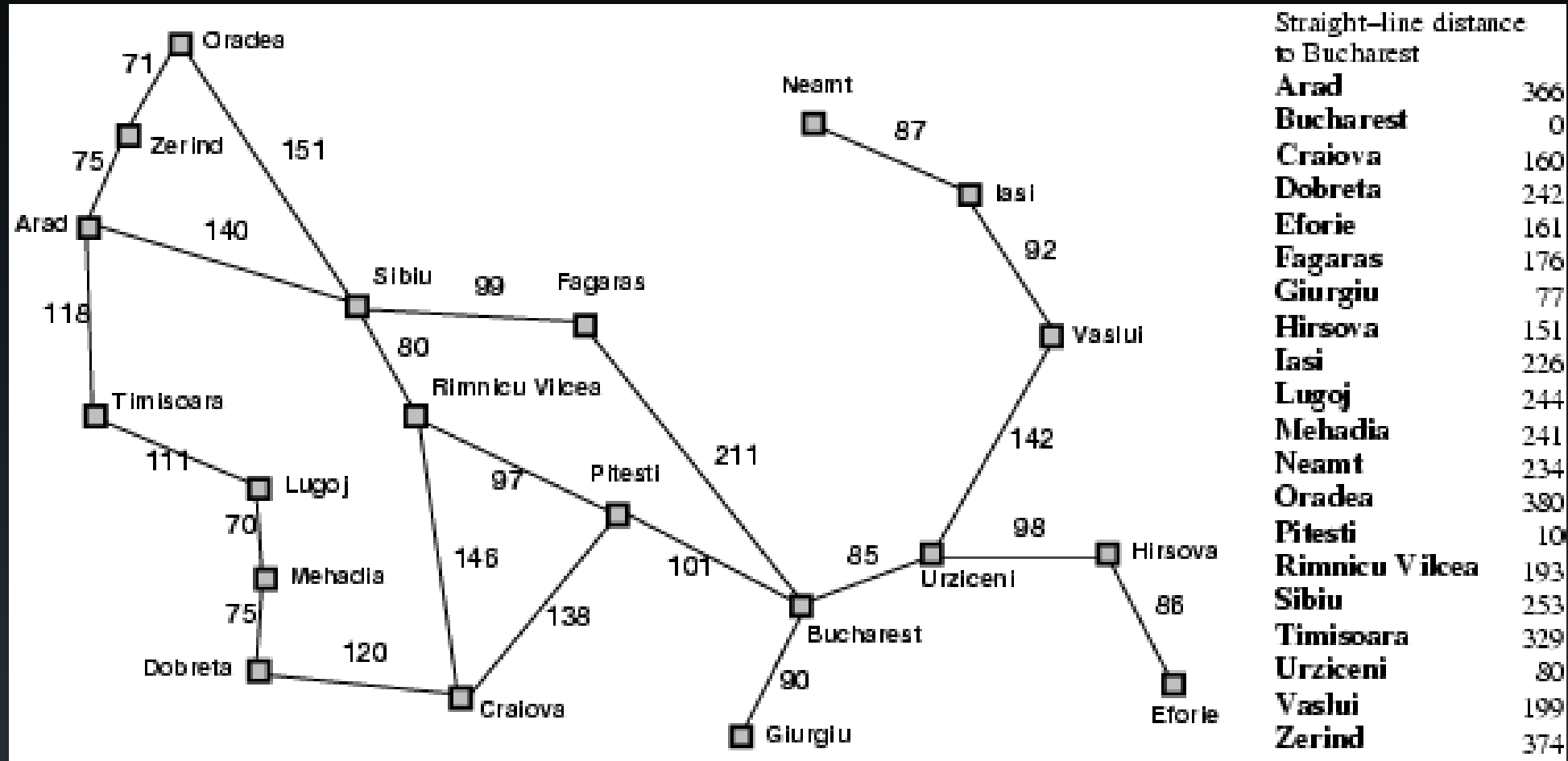
4

- **Rappel** : Une stratégie est définie en choisissant un ordre dans lequel les nœuds sont développés.
- **Idée** : Utiliser une **fonction d'évaluation** f pour chaque nœud
 - mesure l'utilité d'un nœud
 - introduction d'une **fonction heuristique** $h(n)$ qui **estime** le coût du chemin le plus court pour se rendre au but à partir du nœud n .
- La fonction d'évaluation $f(n)$ tente d'estimer le coût du chemin optimal entre le nœud initial et le but, et qui passe par n .
 - en pratique on ne connaît pas ce coût : c'est ce qu'on cherche !
- Insérer le nœud par ordre décroissant d'utilité
- **Cas spéciaux** :
 - La recherche meilleur d'abord gloutonne (greedy best first search).
 - A^*

Exemple d'exploration: Voyage en Roumanie (avec coûts en km)

5

- Chemin entre deux villes = distance Euclidienne (« à vol d'oiseau ») entre la ville n et la ville de destination.



Meilleur d'abord gloutonne

6

l'algorithme meilleur d'abord gloutonne permet d'explorer le nœud le plus proche au but selon une heuristique $h(n)$.

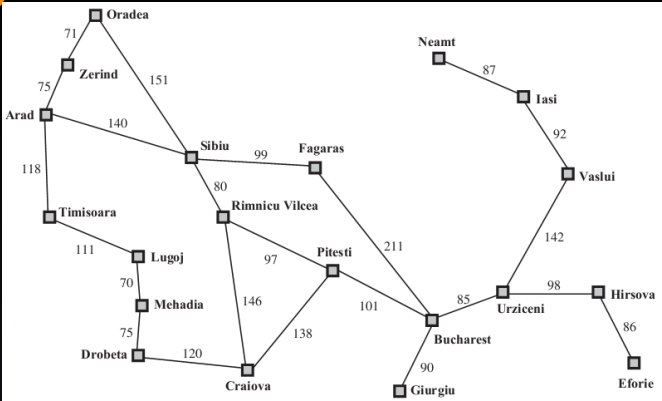
- $f(n) = h(n)$
- Donc on choisit toujours de développer le nœud le plus proche du but.

Exemple meilleur d'abord gloutonne

Frontière

Arad
366

366 Arad

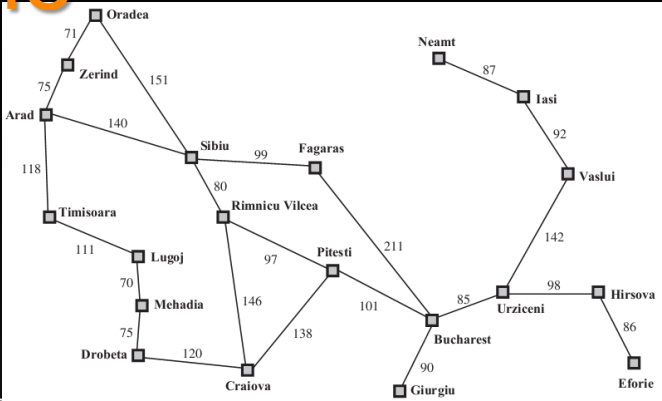
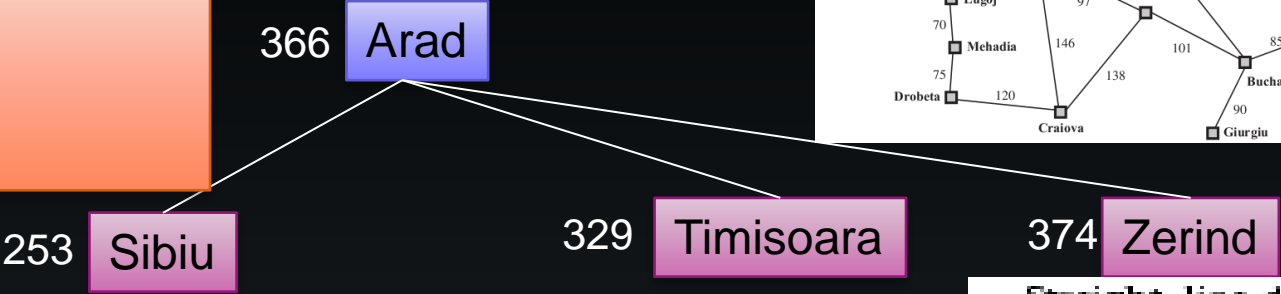


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Exemple meilleur d'abord gloutonne

Frontière

Sibiu	Timisoara	Zerind
253	329	374

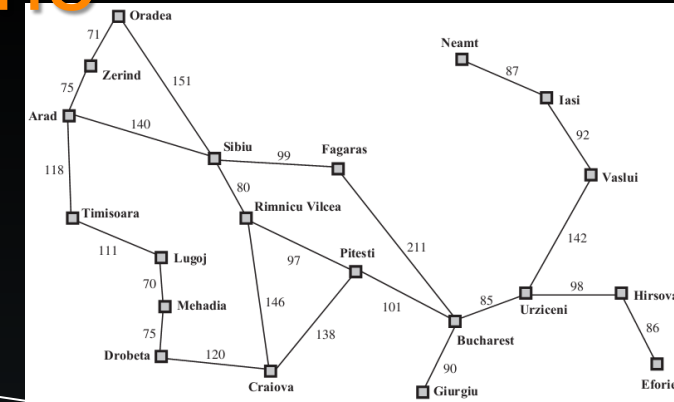
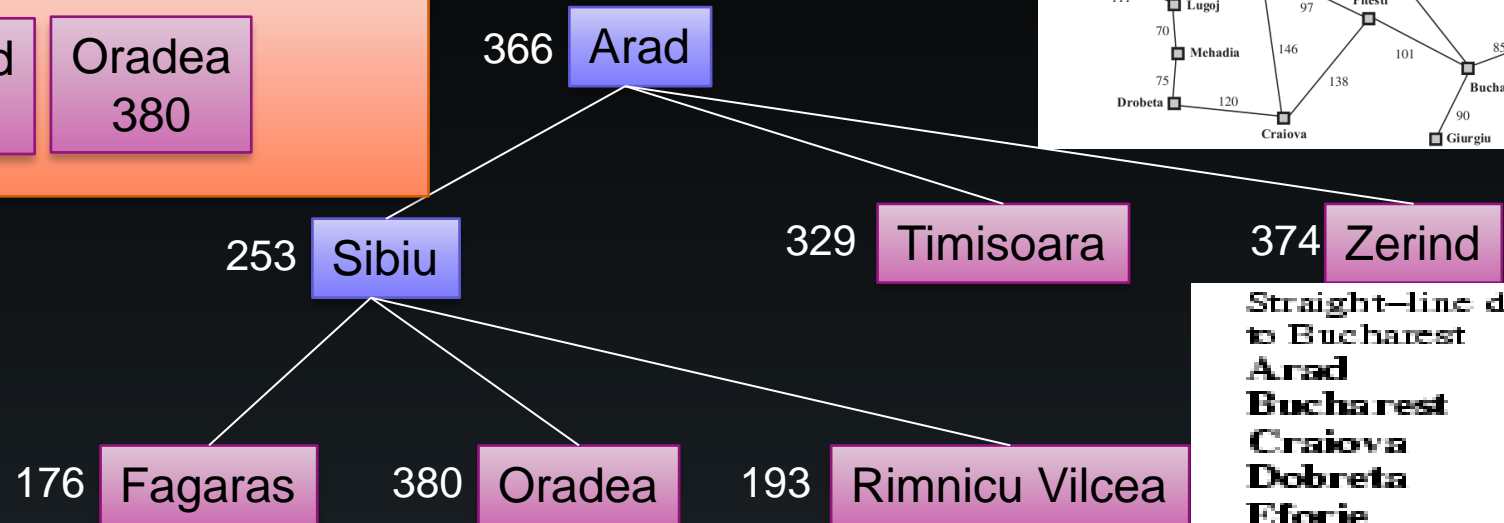


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Exemple meilleur d'abord gloutonne

Frontière

Fagaras 176 Vilcea 193 Timisoara 329 Zerind 374 Oradea 380



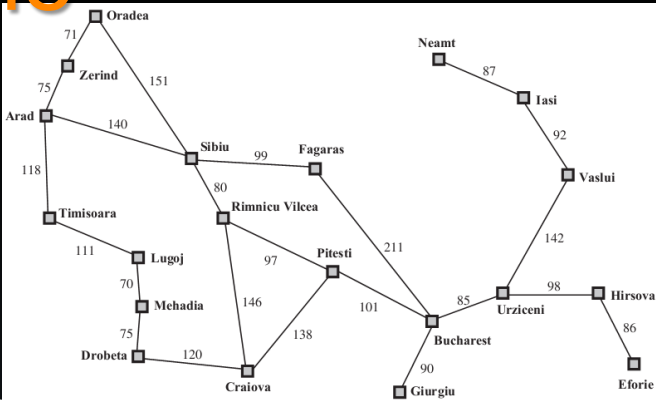
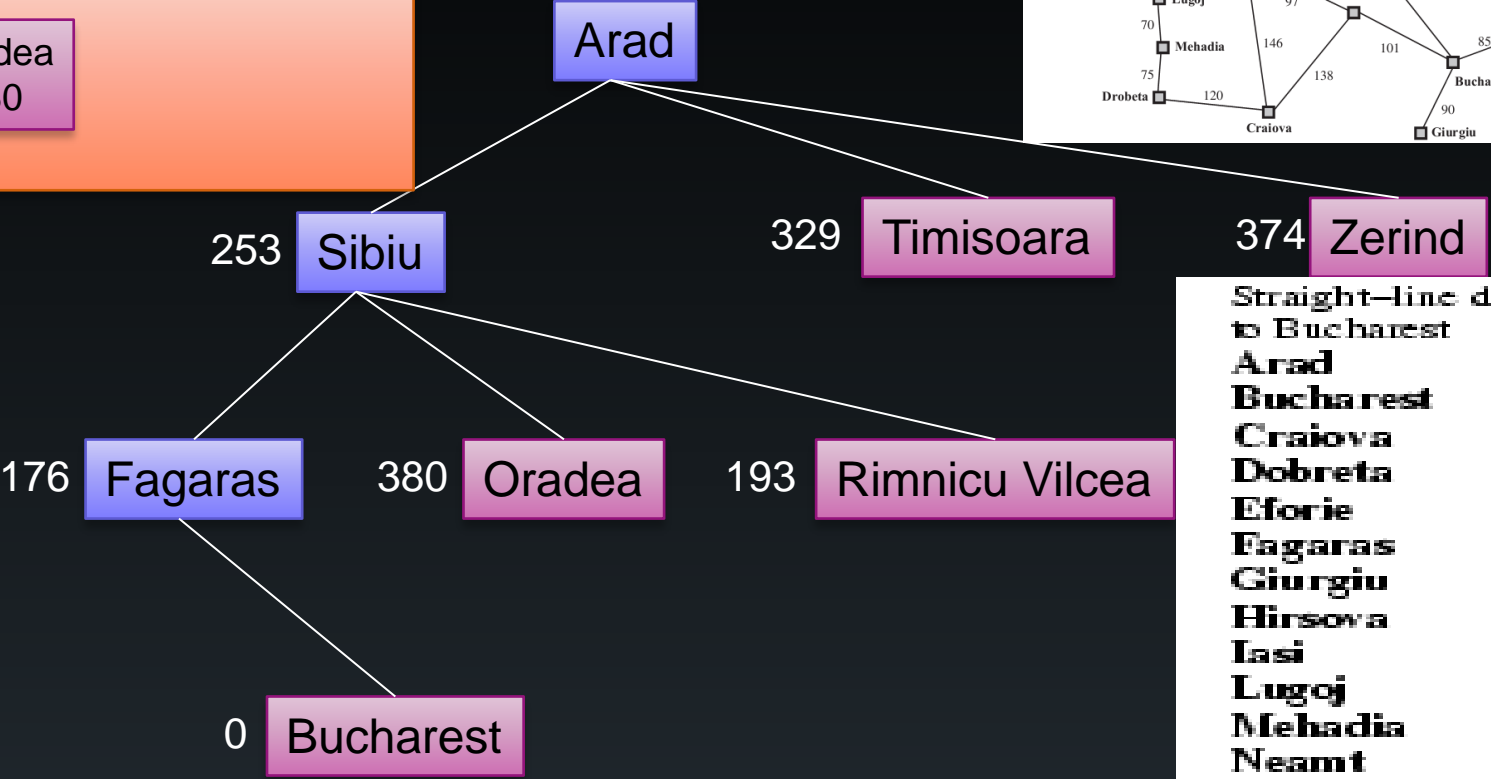
Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Exemple meilleur d'abord gloutonne

Frontière

Bucharest	Vilcea	Timisoara	Zerind	Oradea
0	193	329	374	380

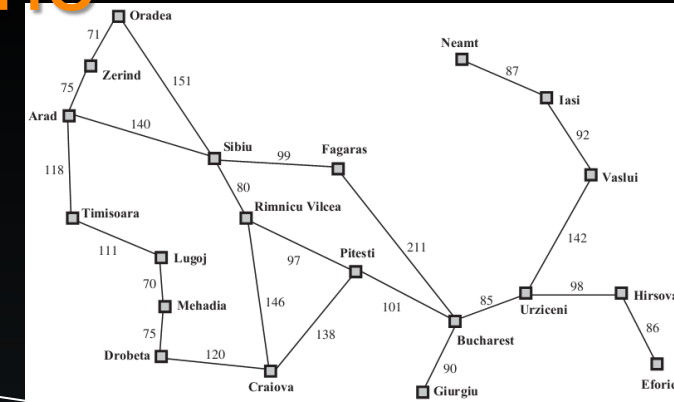
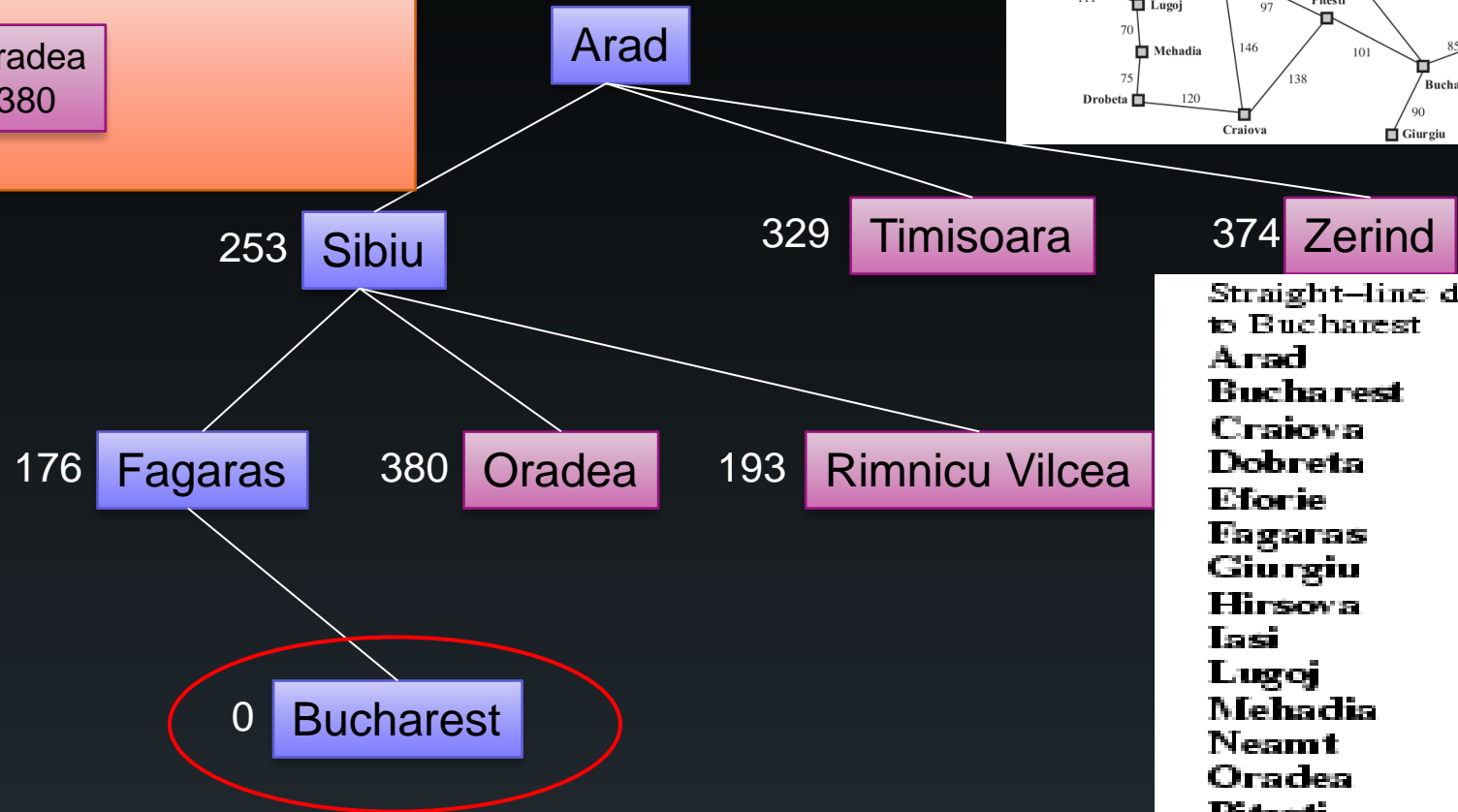


Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Exemple meilleur d'abord gloutonne

Frontière

Vilcea	Timisoara	Zerind	Oradea
193	329	374	380

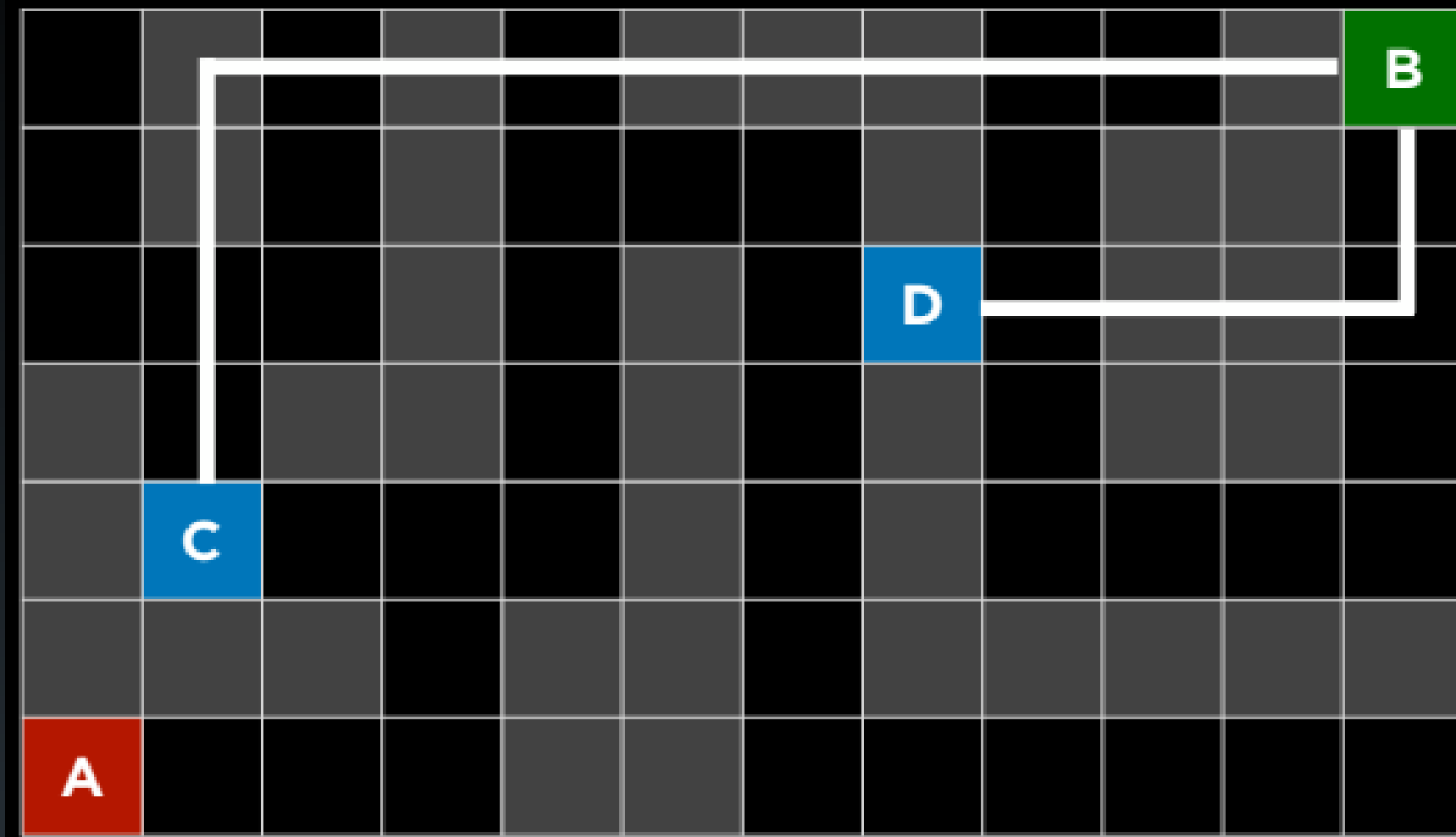


Straight-line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan



Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple2

Fonction heuristique $h(n)$? Distance de Manhattan

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

11		9		7				3	2		B
12		10		8	7	6		4			1
13	12	11		9		7	6	5			2
	13			10		8		6			3
	14	13	12	11		9		7	6	5	4
			13			10					
A	16	15	14			11	10	9	8	7	6

Exemple 3

Fonction heuristique $h(n)$? Distance de Manhattan

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

Propriétés - Meilleur d'abord gloutonne

- Complétude : **Oui**
 - si l'espace de recherche est fini.
- Complexité en temps : $O(b^m)$
 - Mais une bonne fonction heuristique peut améliorer grandement la situation.
- Complexité en espace : $O(b^m)$
 - Elle retient tous les nœuds en mémoire.
- Optimale : **Non**
 - Elle s'arrête à la première solution trouvée.

□ m est la profondeur maximale de l'espace d'états

A* (A-star)

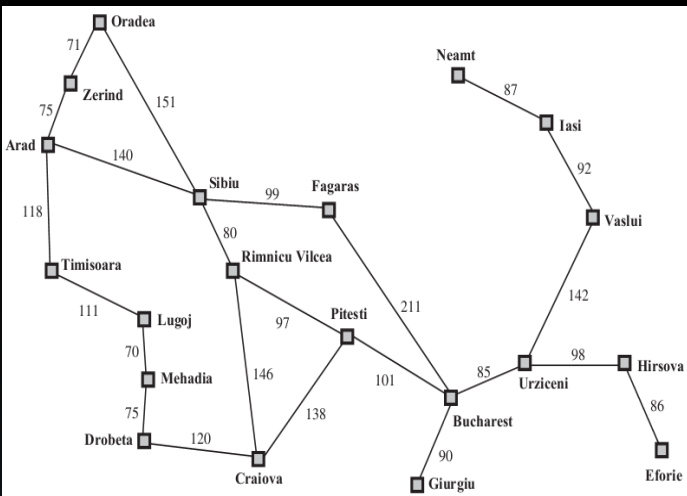
22

- A* utilise une **heuristique**
- Fonction d'évaluation: $f(n) = g(n) + h(n)$
 - $g(n)$: coût pour atteindre le nœud n
 - À tout moment, on connaît seulement le coût optimal pour la partie explorée entre la racine et un nœud déjà exploré.
 - $h(n)$: coût estimé du nœud n jusqu'au nœud but
 - $f(n)$: coût total estimé du chemin passant par n pour se rendre au but.

Example 1

$$f = 0 + 366 = 366$$

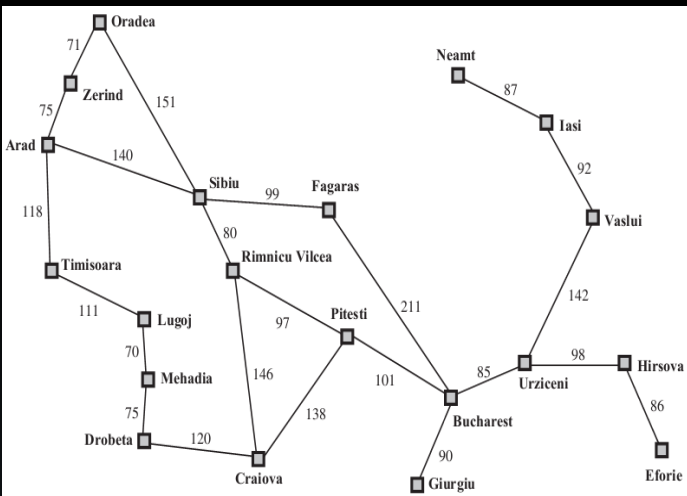
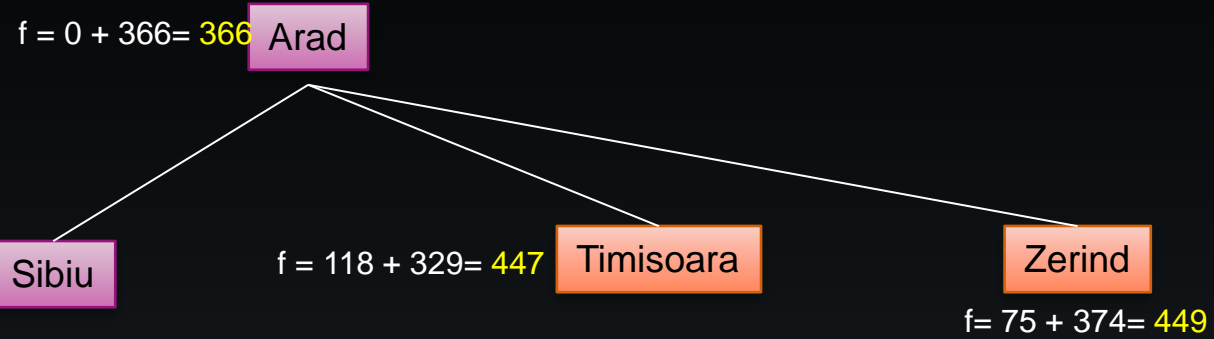
Arad



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	101
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

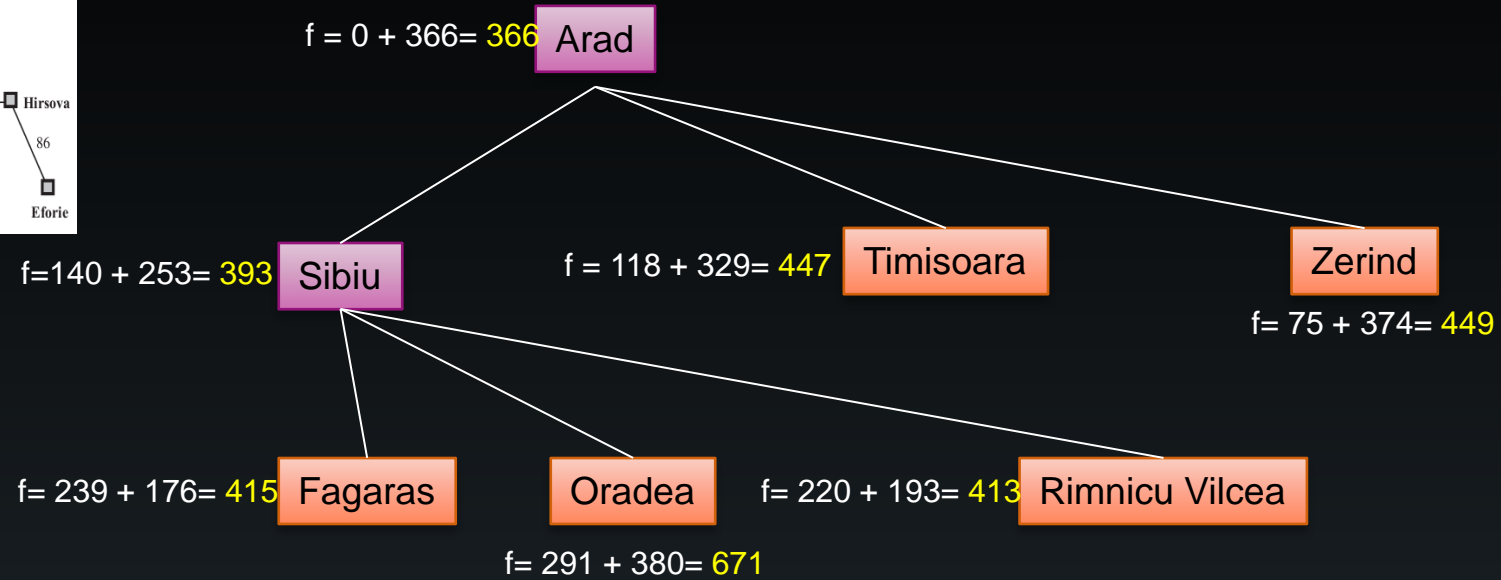
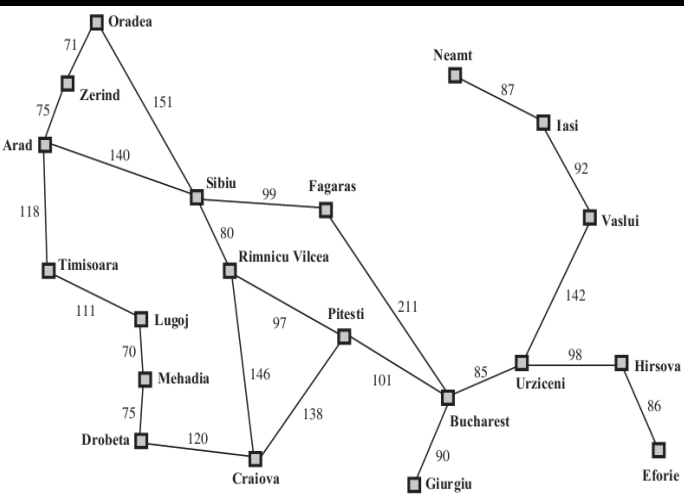
Exemple 1

Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	106
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



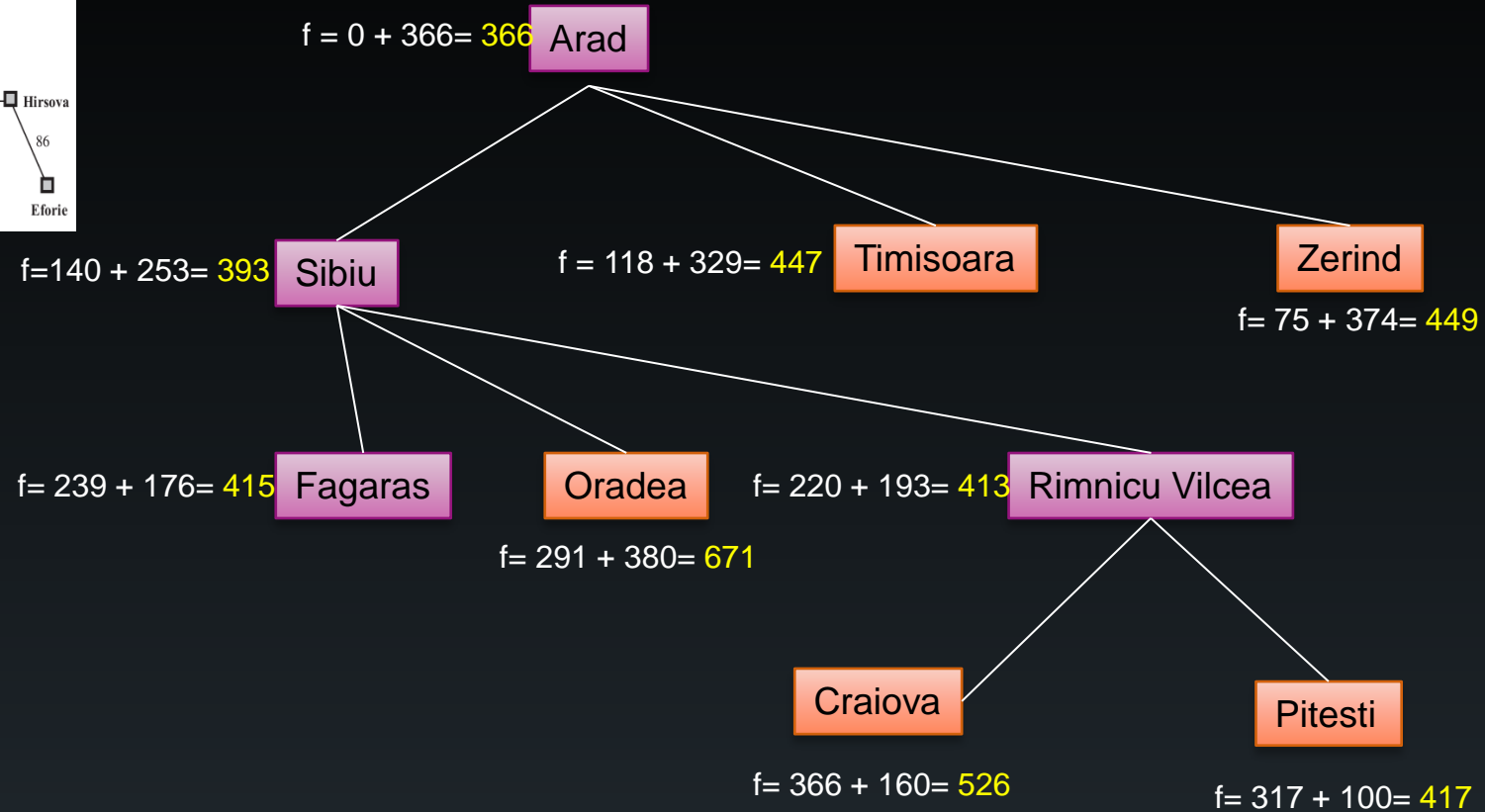
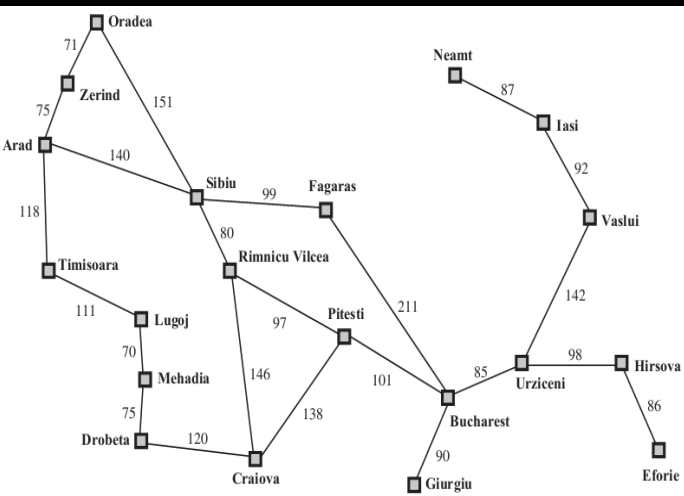
Exemple 1

Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	106
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



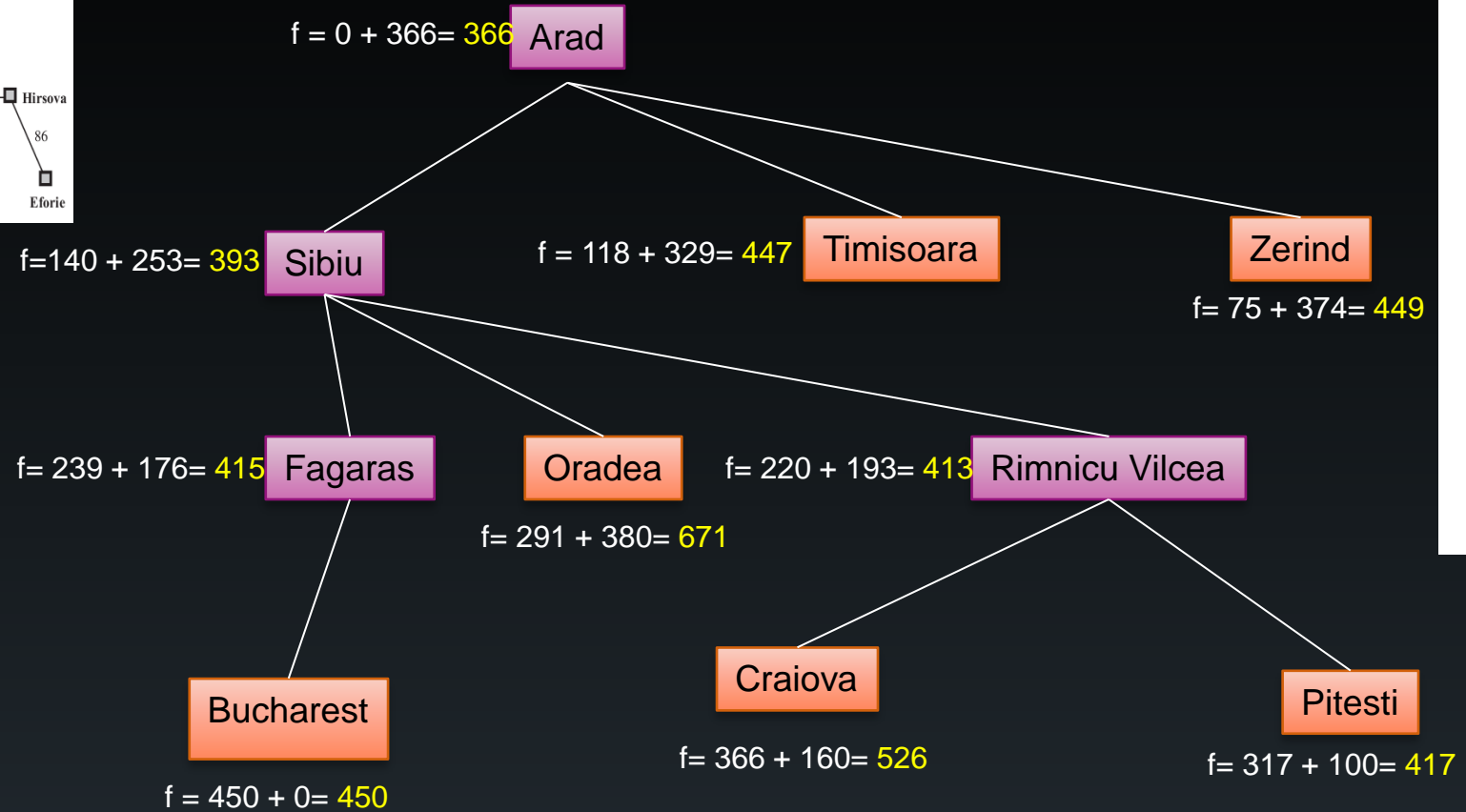
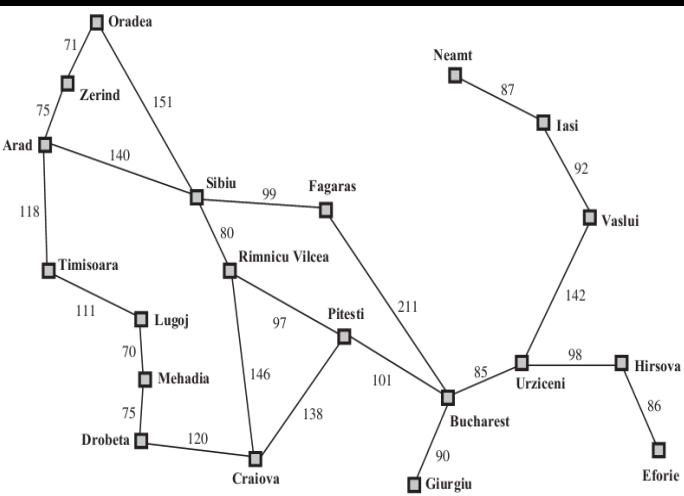
Exemple 1

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



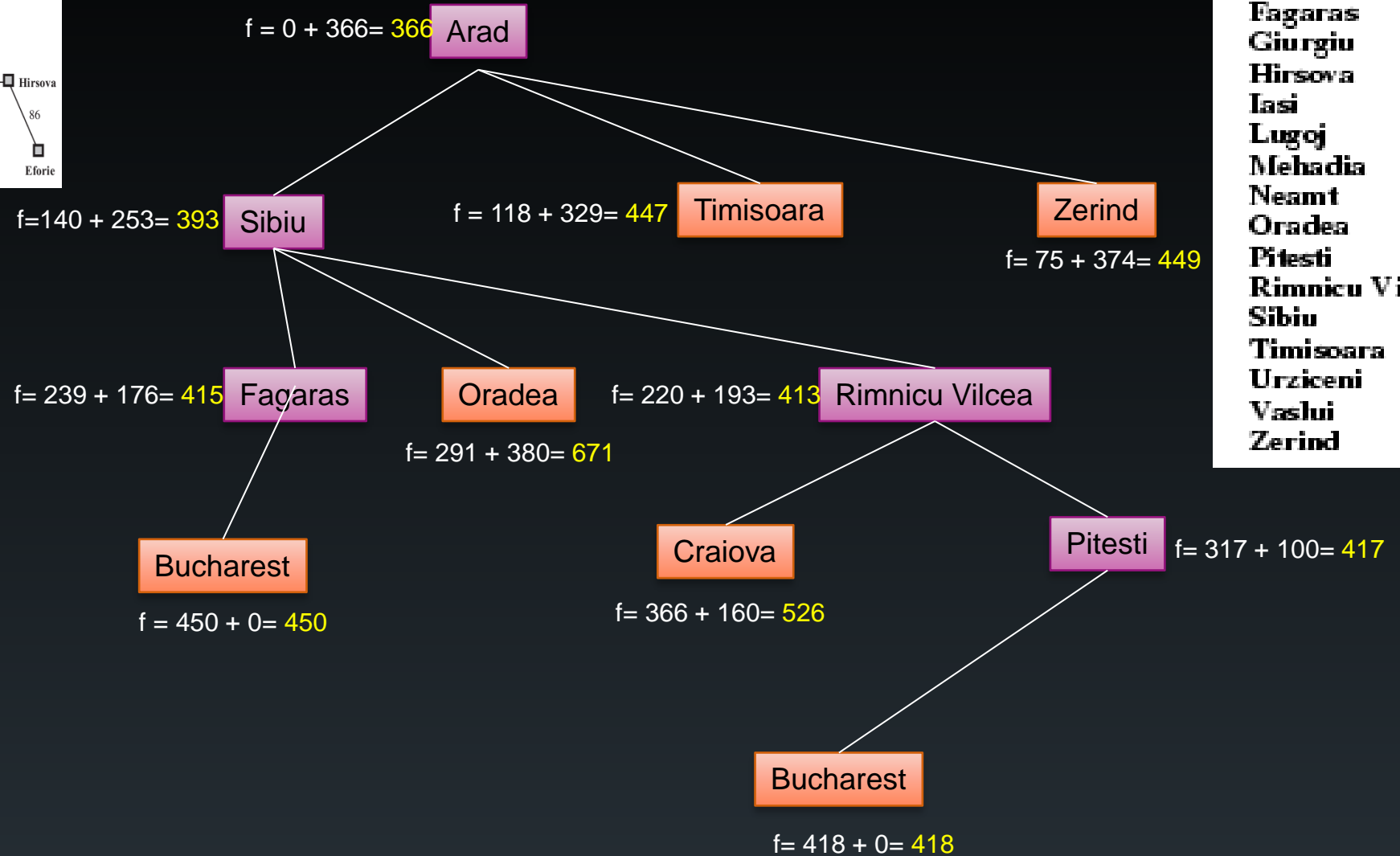
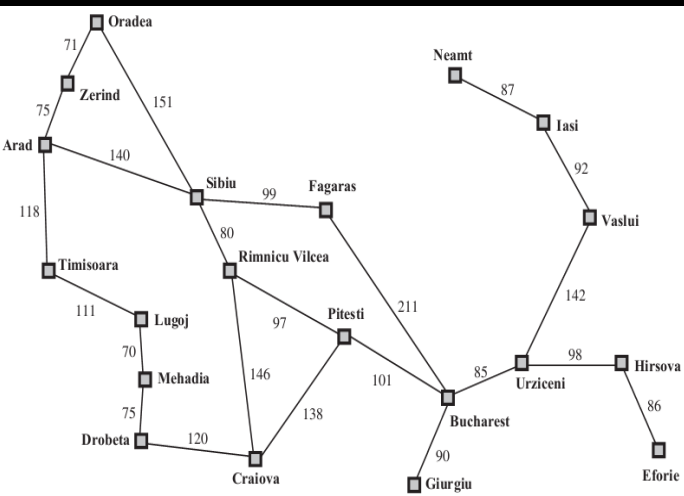
Exemple 1

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



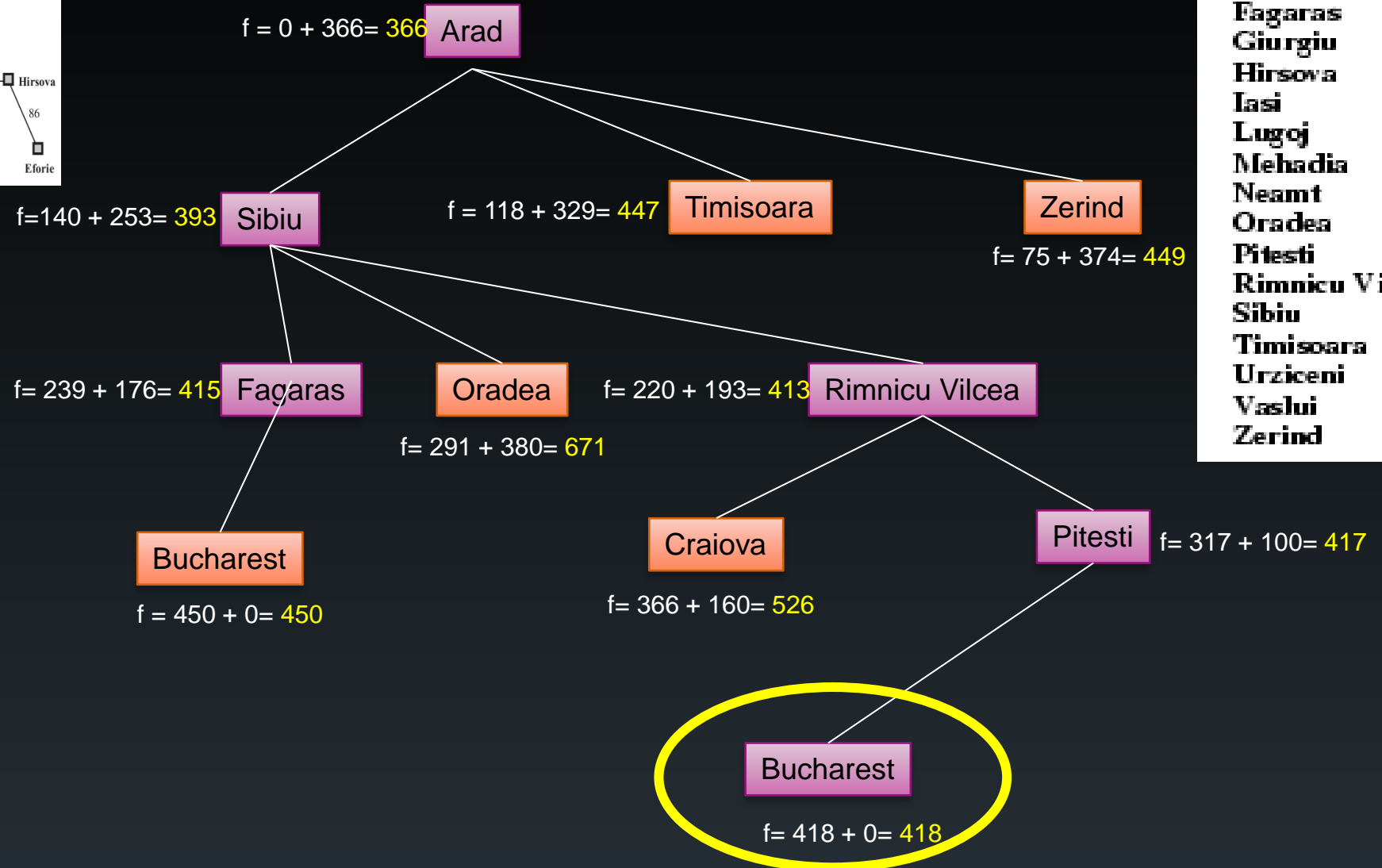
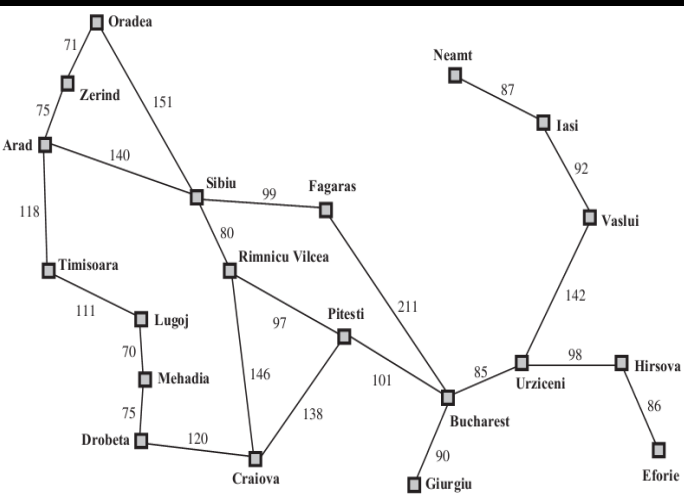
Exemple 1

Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	106
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



Exemple 1

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



Example 2

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	16	15	14		12	11	10	9	8	7	6

Example 2

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	15	14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	2+15	14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		10	9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

Example 2

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

Example 2

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	6+13	5+12		10	9	8	7	6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
	14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	13		6+11						14+5		3
		7+14	6+13	5+12		10	9	8	7	15+6	4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
		8+13		6+11					14+5		3
		7+14	6+13	5+12		10	9	8	7	15+6	4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

Example 2

	10	9	8	7	6	5	4	3	2	1	B
	11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	10	9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

Example 2

	11+10	12+9	13+8	14+7	15+6	5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	15+6	16+5	4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	19+2	1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

	11+10	12+9	13+8	14+7	15+6	16+5	17+4	18+3	19+2	20+1	B
	10+11										1
	9+12		7+10	8+9	9+8	10+7	11+6	12+5	13+4		2
	8+13		6+11						14+5		3
	7+14	6+13	5+12		10	9	8	7	15+6		4
			4+13		11						5
A	1+16	2+15	3+14		12	11	10	9	8	7	6

Propriétés de A*

- Complétude : Oui
 - À moins qu'il y est une infinité de nœuds avec $f \leq f(\text{but})$.
- Complexité de temps : Exponentielle
 - En fonction de la profondeur de la solution.
- Complexité en espace : Exponentielle
 - En fonction de la profondeur de la solution.
 - Elle garde tous les nœuds en mémoire.

Propriétés de A^* : conditions d'optimalité

37

- Si la fonction heuristique h retourne toujours un **estimé inférieur** ou égal au coût réel, on dit que h est **admissible**:
 $h(n) \leq h^*(n)$.
- Dans ce cas, A^* retourne toujours un chemin **optimal**
 - Parfois, on entend par A^* la version de l'algorithme avec la condition additionnelle que h soit **admissible**
 - A^* est alors un **Best-First-Search** où $f(n) = g(n) + h(n)$ et $h(n)$ est **admissible**.

Propriétés de A^*

- Soit $f^*(n)$ le coût **exact** (pas un coût estimé) du chemin optimal du nœud initial au nœud but, passant par n .
- Soit $g^*(n)$ le coût exact du chemin optimal du nœud initial au nœud n .
- Soit $h^*(n)$ le coût exact du chemin optimal du nœud n au nœud but.
- On a donc que $f^*(n) = g^*(n) + h^*(n)$.
- Si l'heuristique est **admissible**, pour chaque nœud n exploré par A^* , on peut montrer que l'on a toujours
 - $f(n) \leq f^*(n)$.

Propriétés de A*

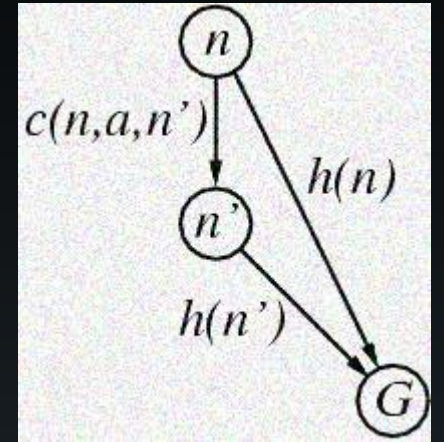
- On dit que h est **consistante (cohérente)**; si quelque soit un nœud n et son successeur n' , nous avons.

$$h(n) \leq c(n, n') + h(n')$$

où $c(n, n')$ est le coût de l'arc (n, n') .

Dans ce cas :

- ❖ h est aussi **admissible**
- ❖ chaque fois que A* choisit un nœud au début de OPEN, cela veut dire que A* a déjà trouvé un chemin **optimal** vers ce nœud :
 - ❖ le nœud ne sera plus jamais revisité!



Propriétés de A*

- Si on a deux heuristiques admissibles h_1 et h_2 , tel que $h_1(n) < h_2(n)$, alors $h_2(n)$ conduit **plus vite** au but
 - En utilisant **h_2** , A* explore **moins** ou autant de nœuds avant d'arriver au but qu'en utilisant **h_1** .

Exploration heuristique à mémoire limitée

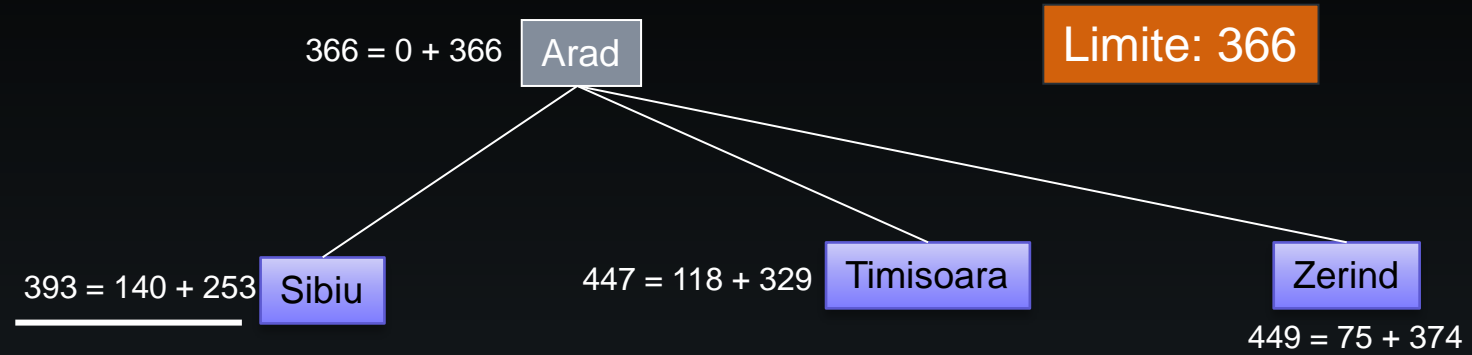
- A* est parfois trop gourmand en mémoire.
- Il existe des algorithmes pour surmonter ce problème dont:
 - IDA*;
- Ces algorithmes permettent de préserver l'optimalité et la complétude.
- L'augmentation du temps d'exécution est raisonnable.

Exploration heuristique à mémoire limitée

- IDA* (*Iterative-deepening A**)
 - C'est un algorithme de profondeur itérative
 - Utilise la valeur $f(n)$ comme limite
 - À chaque itération :
 - On fixe la limite à la plus petite valeur $f(n)$ de tous les nœuds qui avaient une valeur plus grande que la limite au tour précédent.

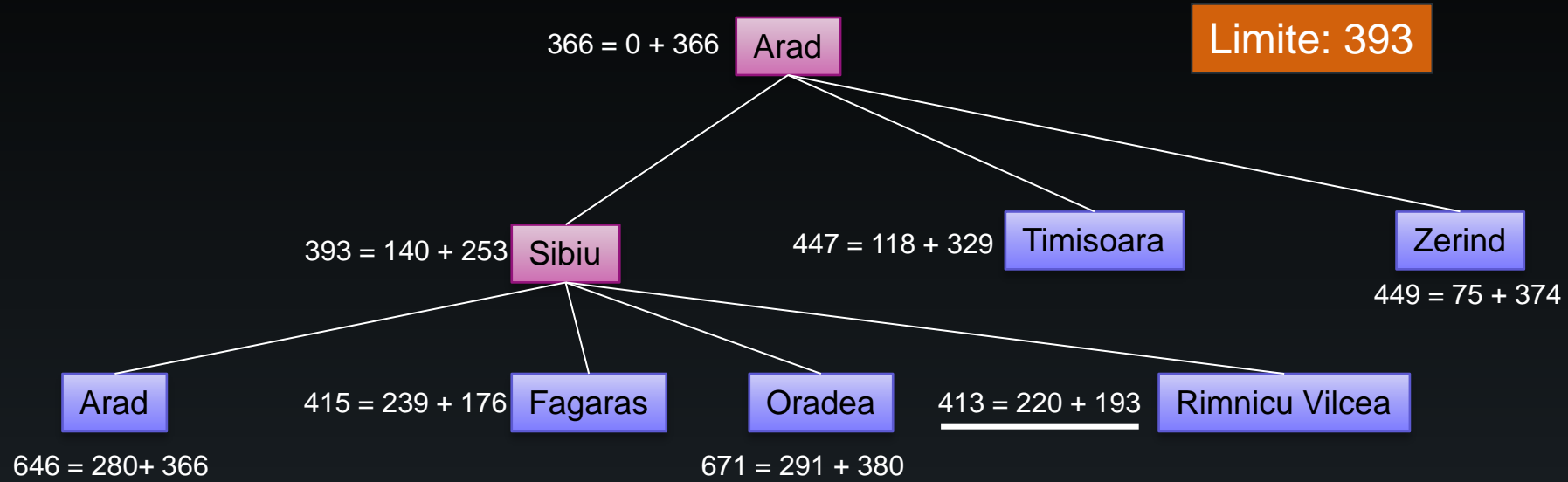
Exemple IDA*

43



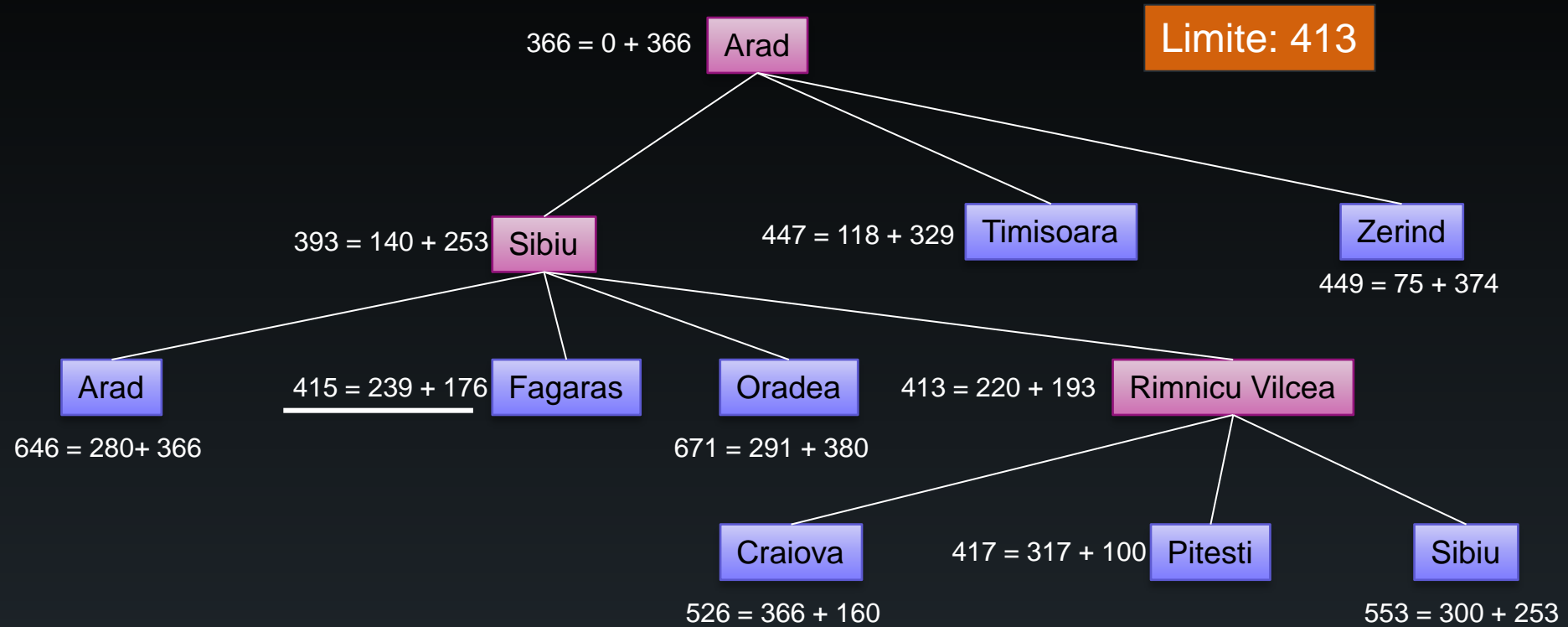
Exemple IDA*

44



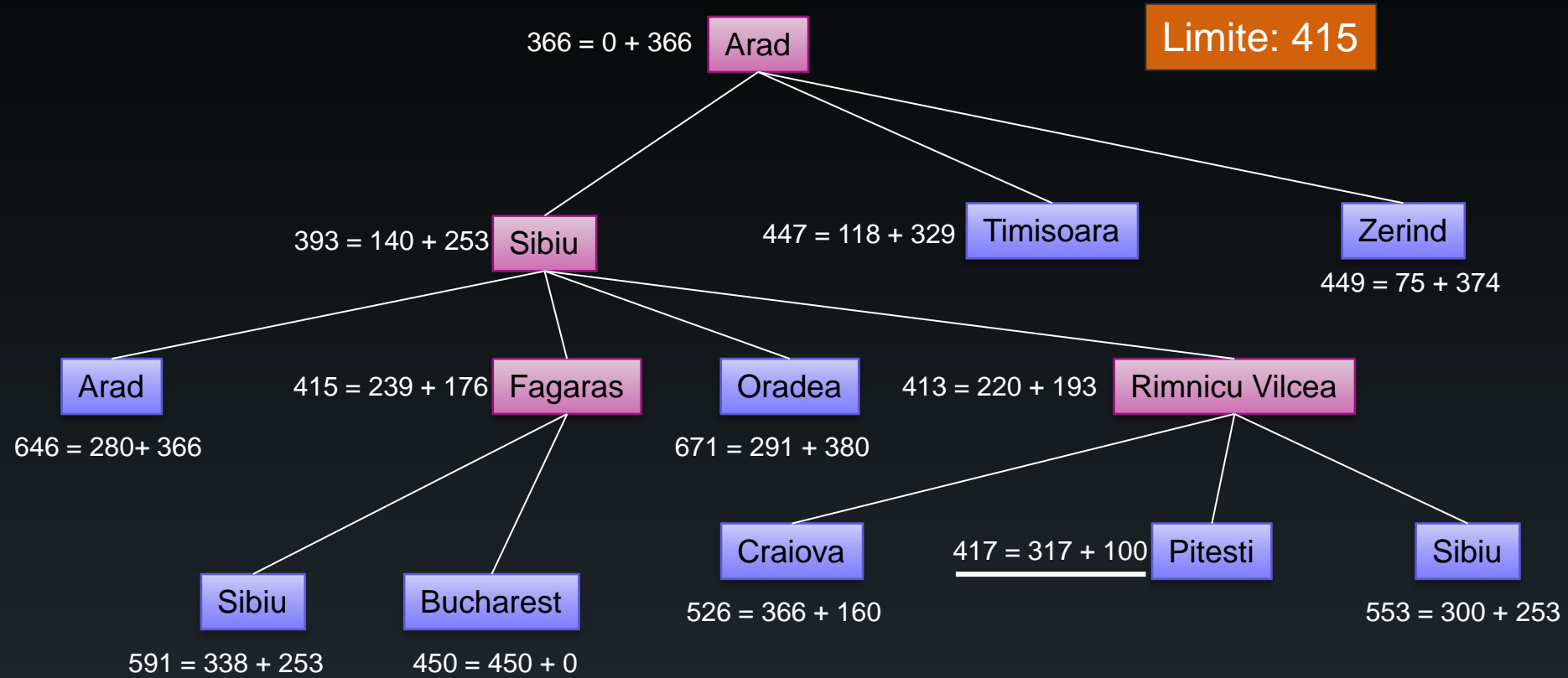
Exemple IDA*

45



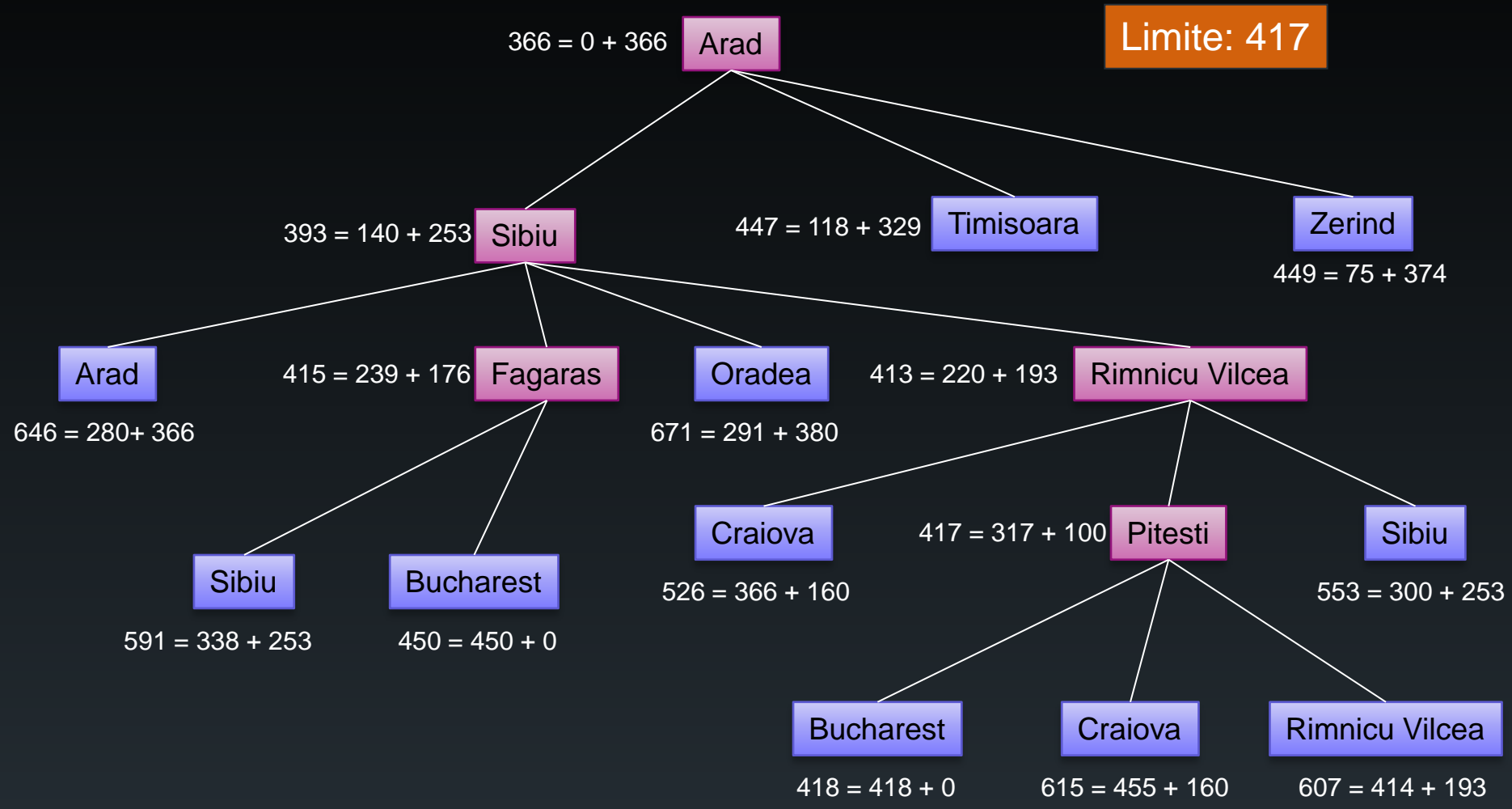
Exemple IDA*

46



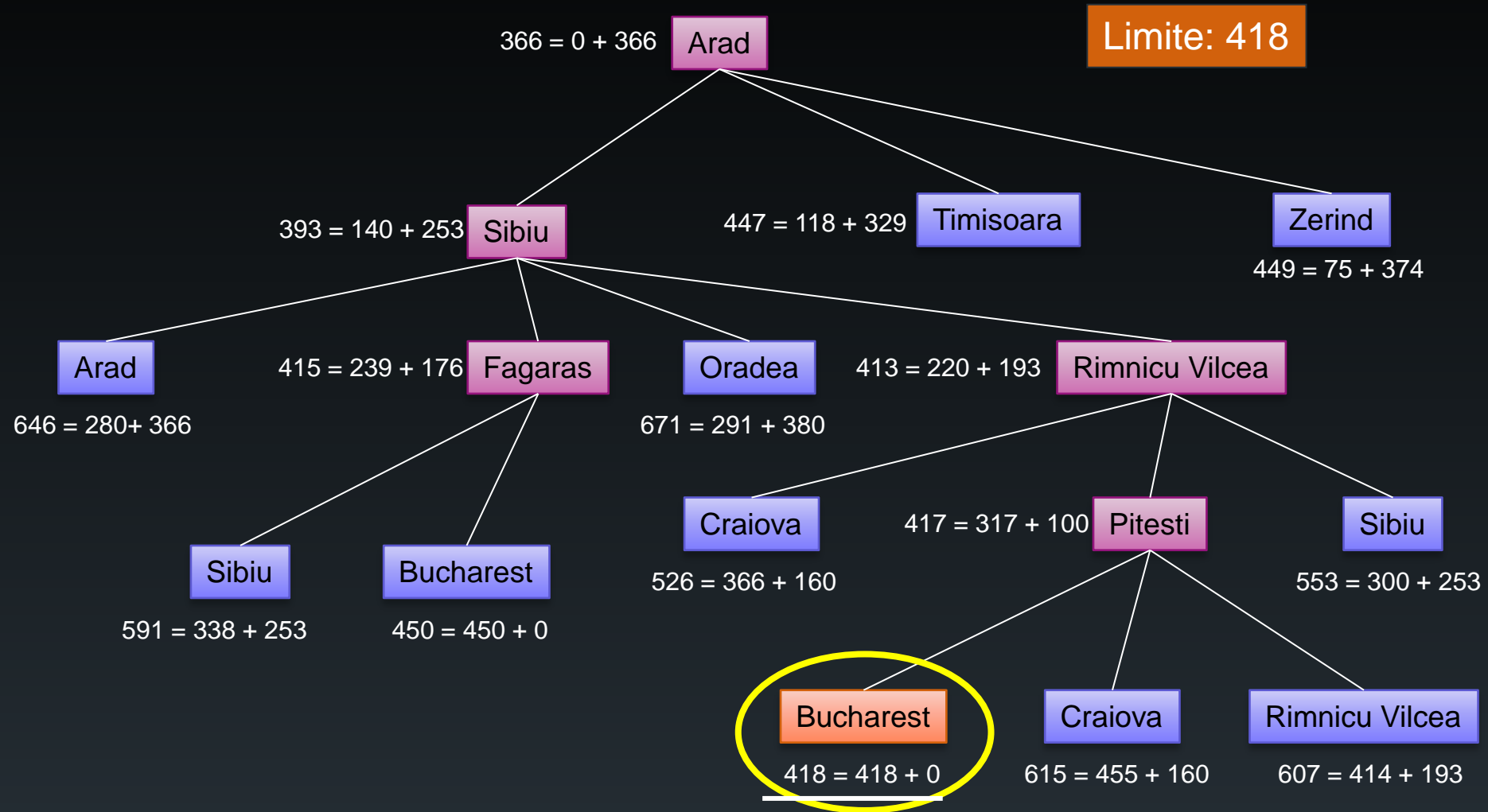
Exemple IDA*

47



Exemple IDA*

48



Fonctions heuristiques

49

7	2	4									
5		6									
8	3	1									

État initial

	1	2									
3	4	5									
6	7	8									

État but

❑ Le nombre d'états possibles pour un problème de taquin à 8 généré aléatoirement est :

$$\frac{9!}{2} = 181440$$

❑ Pour le problème de taquin à 15, ce nombre devient 10^{13}

Fonctions heuristiques

50

7	2	4						
5		6						
8	3	1						

État initial

	1	2						
3	4	5						
6	7	8						

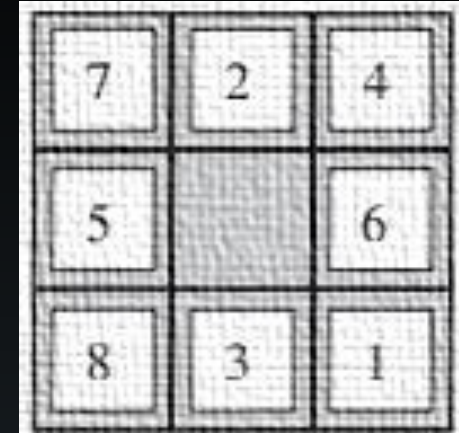
État but

- ❑ Utiliser une heuristique pour résoudre le problème d'une manière efficace
- ❑ Les deux heuristiques les plus utilisées pour résoudre le jeu de taquin à 15 sont :
 - ❑ H1 : le nombre de pièces mal placées. Pour l'état initial $h1=8$ puisque toutes les pièces sont mal placées.
 - ❑ H1 est admissible : chaque pièce mal placée doit être déplacée au moins une seule fois.
 - ❑ H2 : la somme des distances entre chaque pièce et sa position dans l'état but. Pour l'état initial $h2= 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$
 - ❑ H2 est admissible.

Génération des heuristiques admissibles à partir de problèmes relaxés

51

- ❑ Simplifier (ou relaxer) le problème
- ❑ Taquin à 8 :
 - ❑ Une pièce peut être déplacée de A à B si A est adjacente horizontalement ou verticalement à B et B est vide
- ❑ On peut générer plusieurs versions simplifiées en supprimant l'une des deux conditions.
 - 1) A peut être déplacée à B si A et B sont adjacentes
 - 2) A peut être déplacée à B si B est vide
 - 3) On peut déplacer une pièce A à B
- ❑ À partir de la première version simplifiée on peut dériver l'heuristique h2
- ❑ À partir de la version 3 on peut dériver h1



Génération automatique des heuristiques

52

- En testant plusieurs simplifications du problèmes.
- Si on trouve plusieurs heuristiques et qu'il n'y en a pas une qui en domine une autre, alors on peut utiliser l'heuristique composite :
 - $h(n) = \max\{h_1(n), \dots, h_m(n)\}$