

AI 2002

Artificial Intelligence

Dr. Hashim Yasin

Informed (Heuristic) Search Strategies

Informed Search

- ▶ One that uses **problem-specific knowledge** beyond the definition of the problem itself.
- ▶ For Example
 - ▶ Best First Search
 - ▶ A* Search

Best-First Search

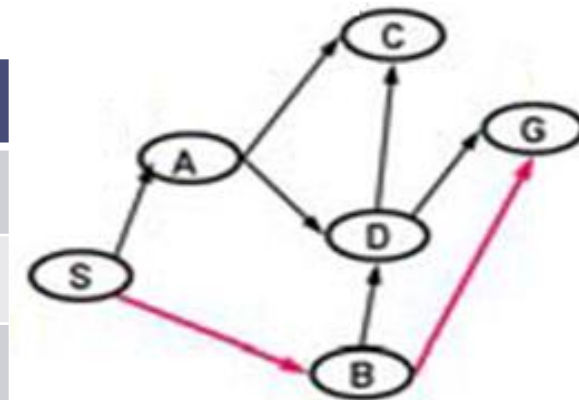
- ▶ A node is selected for expansion based on an **evaluation function**, $f(n)$.
 - the *lowest* evaluation is expanded first
- ▶ Best-first search algorithms include as a component of f a **heuristic function**, denoted $h(n)$.
- ▶ $h(n)$ = estimated cost of the cheapest path
 - $h(n)$ takes a *node* as **input**, but, unlike $g(n)$, it depends only on the *state* at that node.

$$f(n) = h(n)$$

Best-First Search

- ❑ Pick best (by **heuristic value**) element of Q
- ❑ Add path extensions to Q

	Q	Visited
1	(10 S)	S
2	(2 A S) (3 B S)	A, B, S
3	(1 C A S) (3 B S) (4 D A S)	C, D, B, A, S
4	(3 B S) (4 D A S)	C, D, B, A, S
5	(0 G B S) (4 D A S)	G, C, D, B, A, S



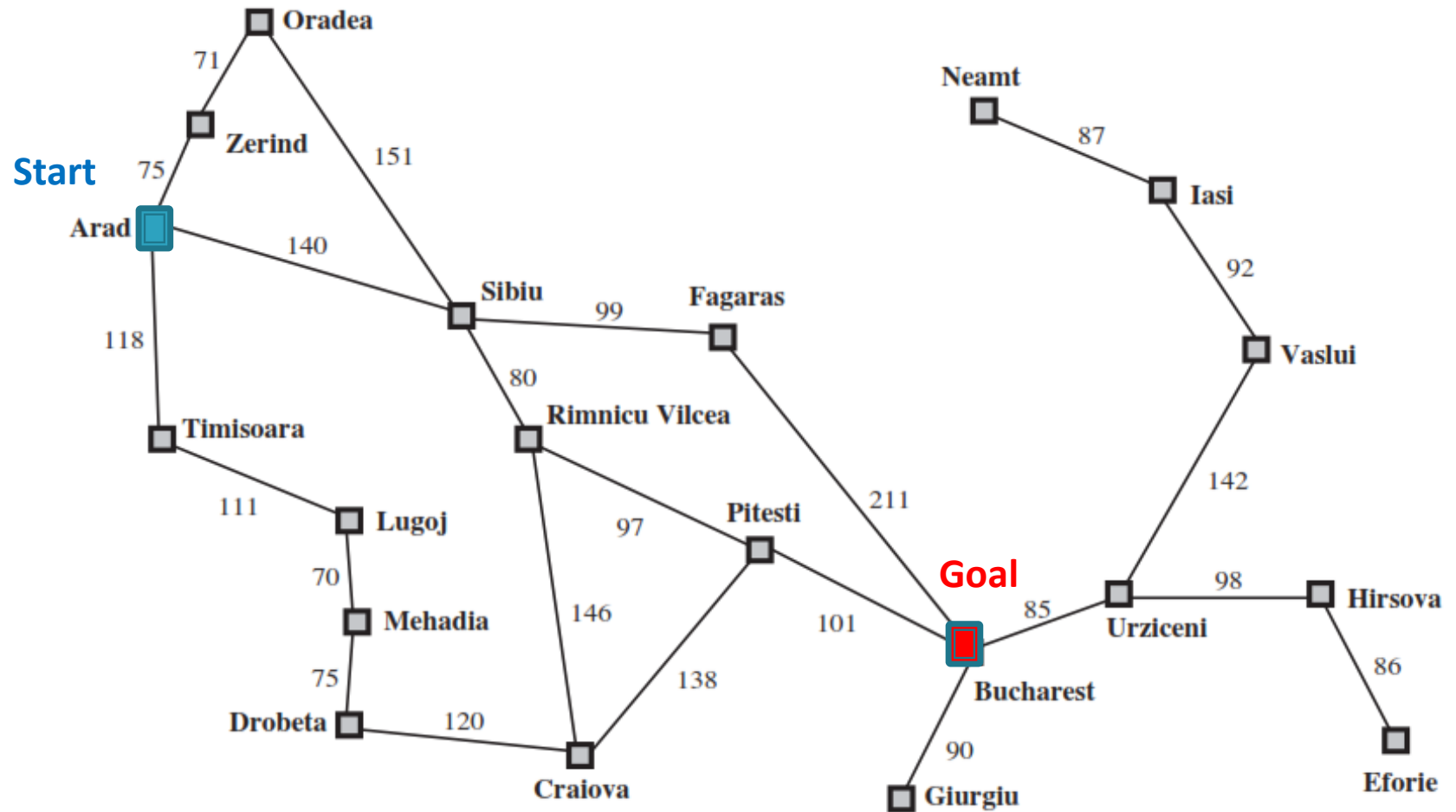
Heuristic Values

A=2 C=1 S=10
B=3 D=4 G=0

- ❑ Blue Color represents added paths
- ❑ Heuristic value in node state is in front.

(4DBS) is not included in the list because 'D' is already available in "Visited List"

Example



A simplified road map of part of Romania.

Example

Values of h_{SLD} —straight-line distances to Bucharest

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
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

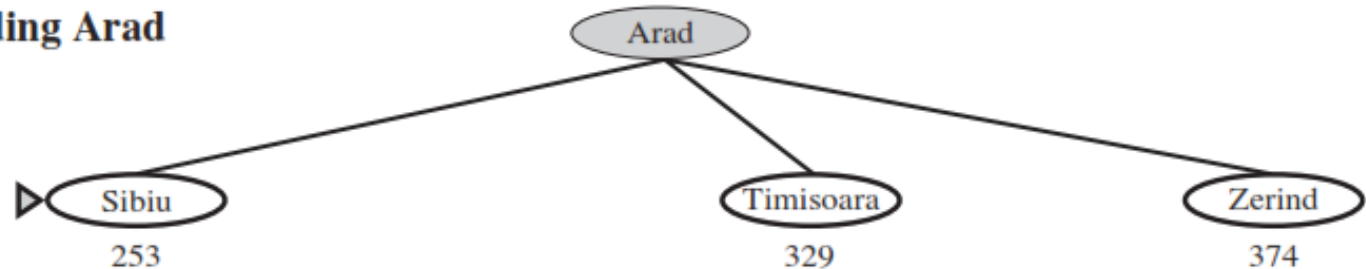
Best-First Search

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
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

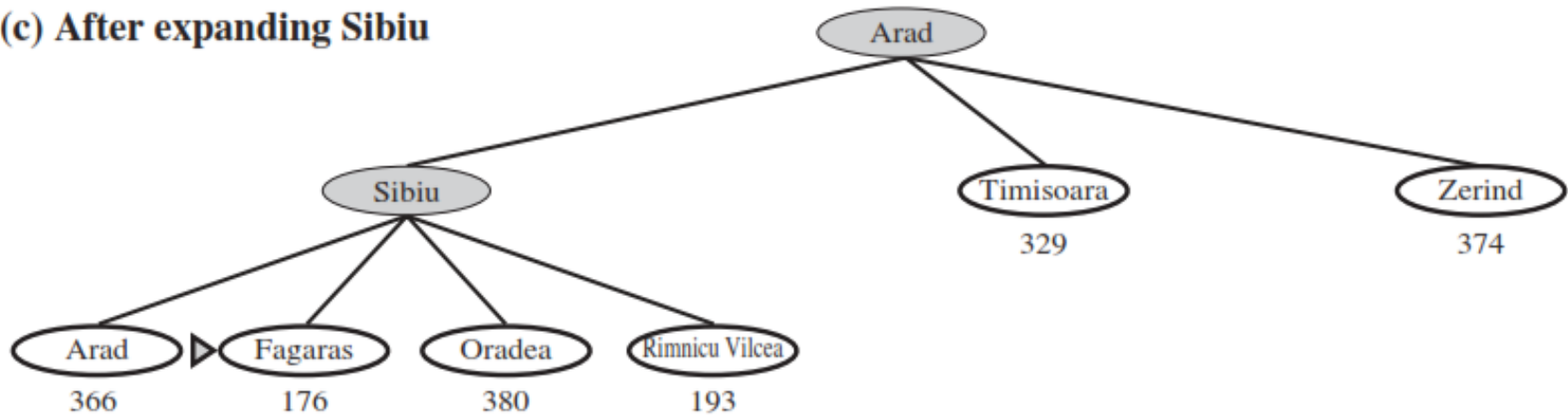
(a) The initial state



(b) After expanding Arad



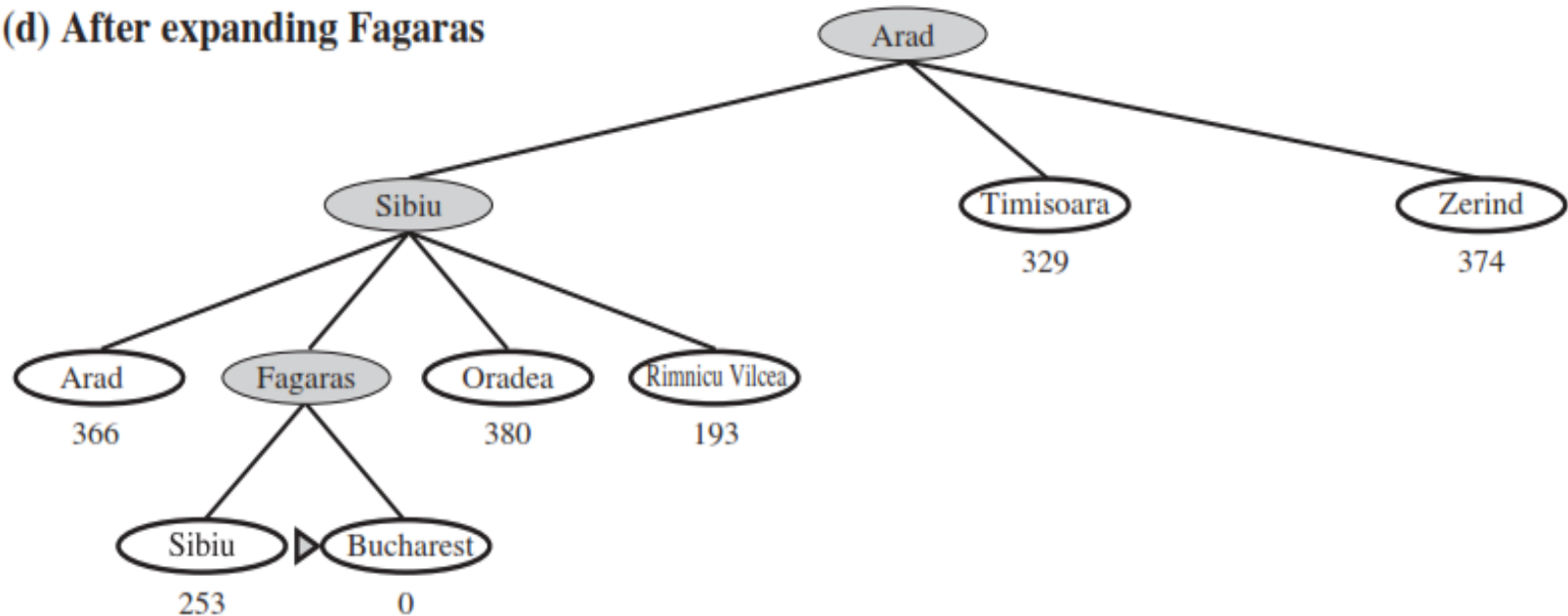
(c) After expanding Sibiu



Best-First Search

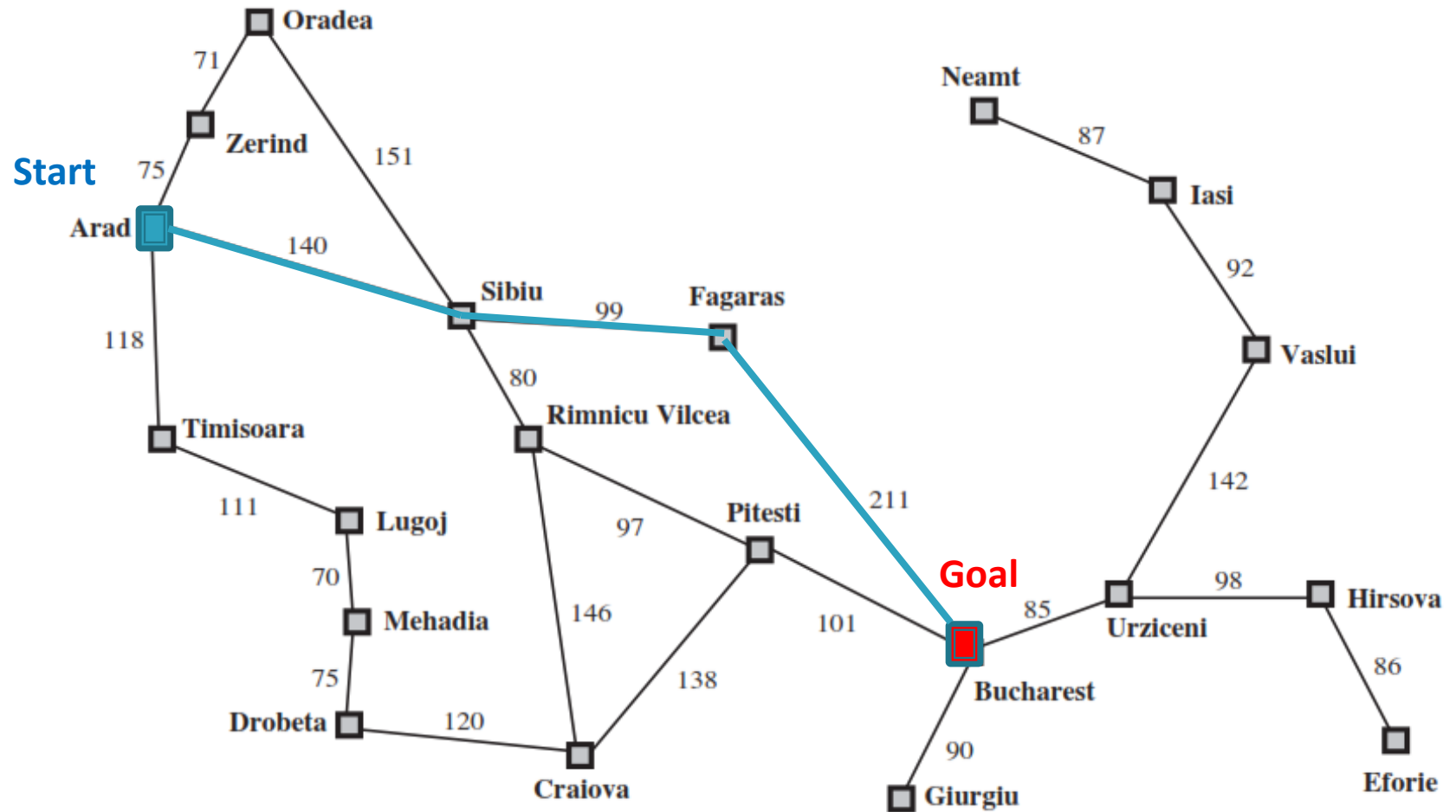
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

(d) After expanding Fagaras



Stages in a greedy best-first tree search for Bucharest with the **straight-line distance heuristic**. Nodes are labeled with their **h-values**.

Example



A simplified road map of part of Romania.

Best-First Search

Completeness

- ▶ Best-first **tree search** is **incomplete** even in a finite state space, much like depth first search.
- ▶ The **graph search** version is **complete** in finite spaces, but not in infinite ones.

Time and Space Complexity

- ▶ The *worst-case time and space* complexity for the tree version is $O(b^m)$, where m is the maximum depth of the search space.
 - With a good heuristic function, however, the complexity can be reduced substantially.

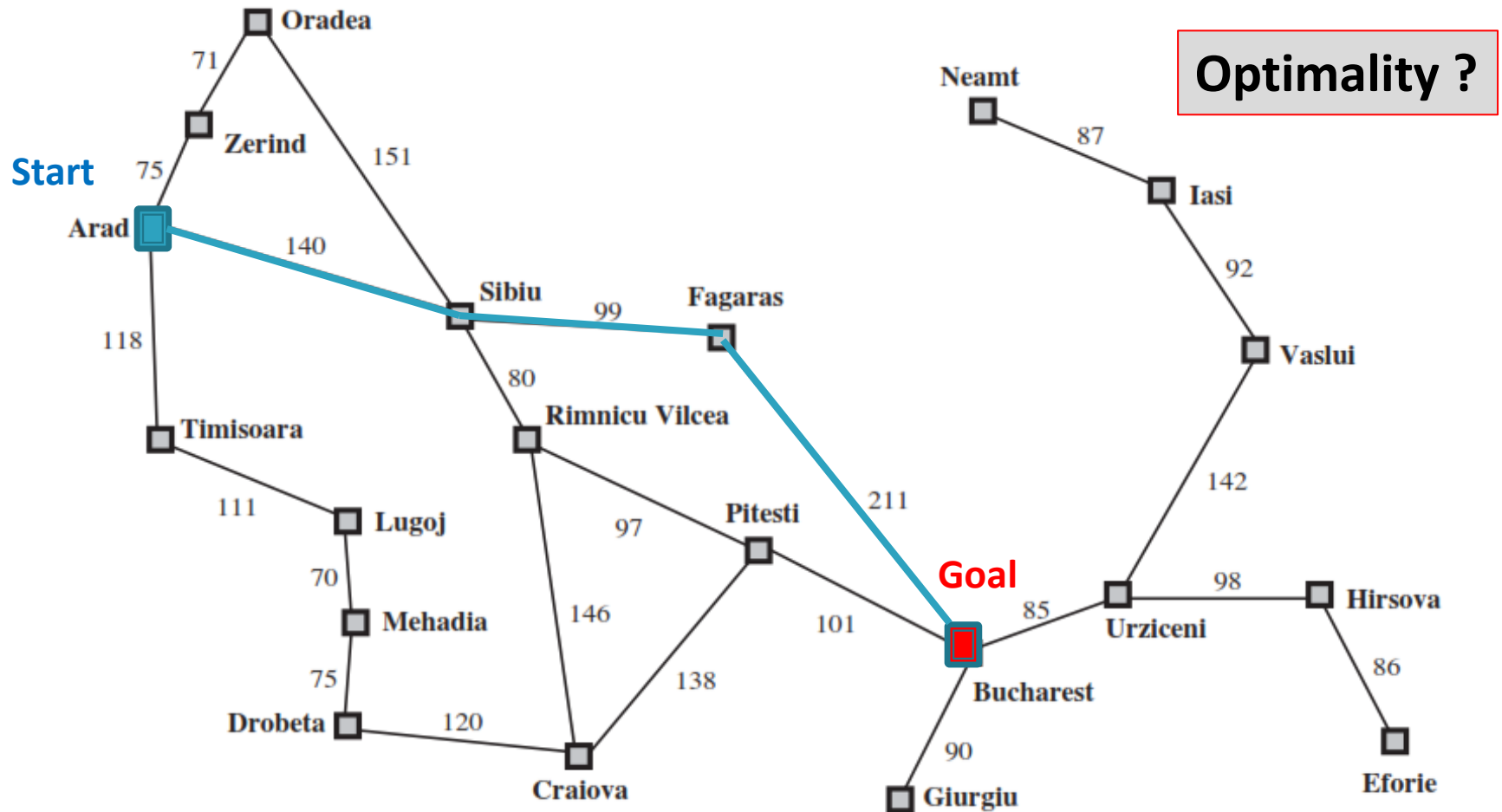
Example 1

Arad-Sibiu-Fagaras-Buchrest:

$140+99+211=450$

Arad-Sibiu-Vilcea-Pitesti-Buchrest:

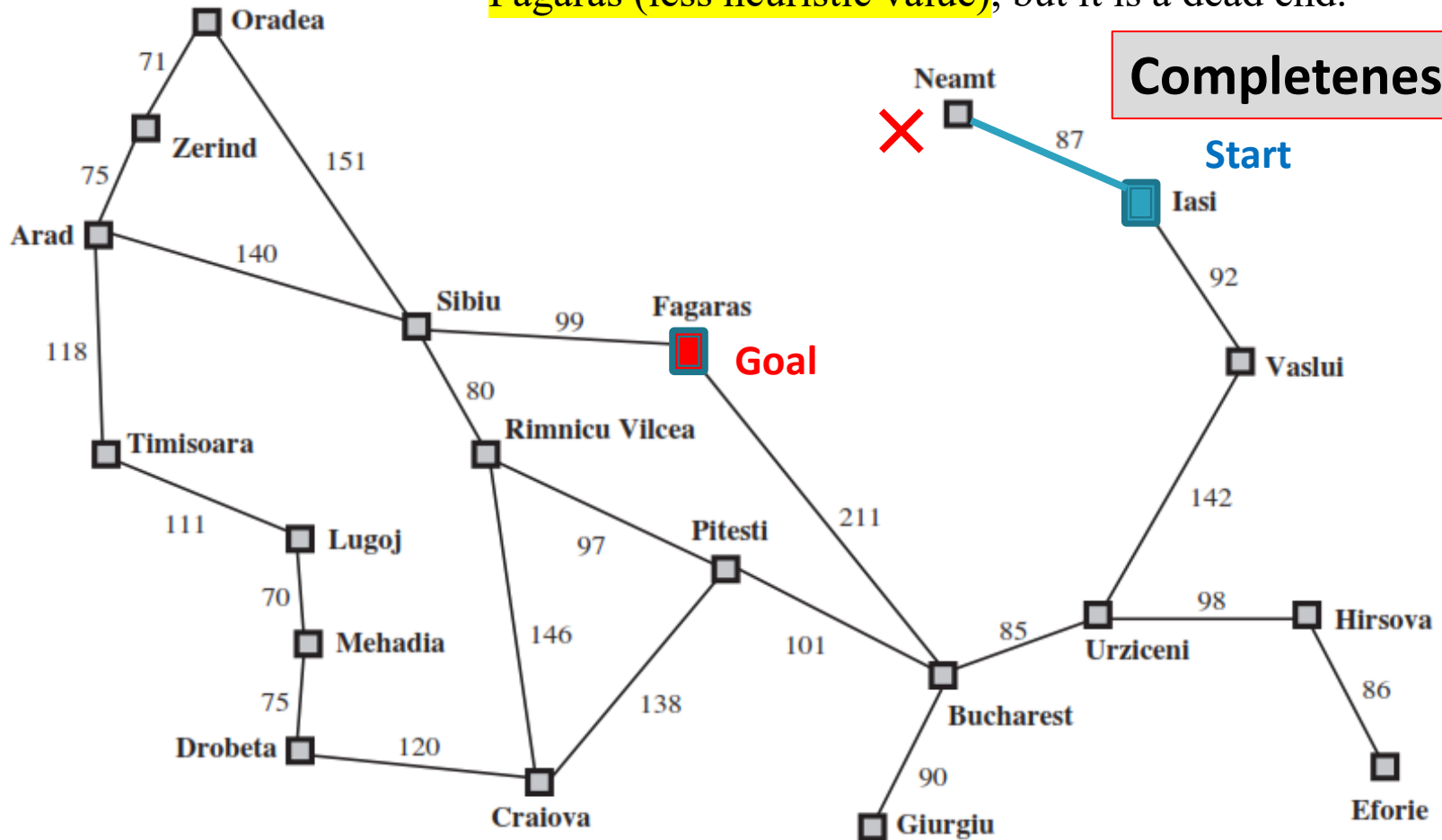
$140+80+97+101=418$



A simplified road map of part of Romania.

Example 2

The heuristic--- **straight-line distances** ---suggests that **Neamt** be expanded first because it is closest to **Fagaras** (less heuristic value), but it is a dead end.

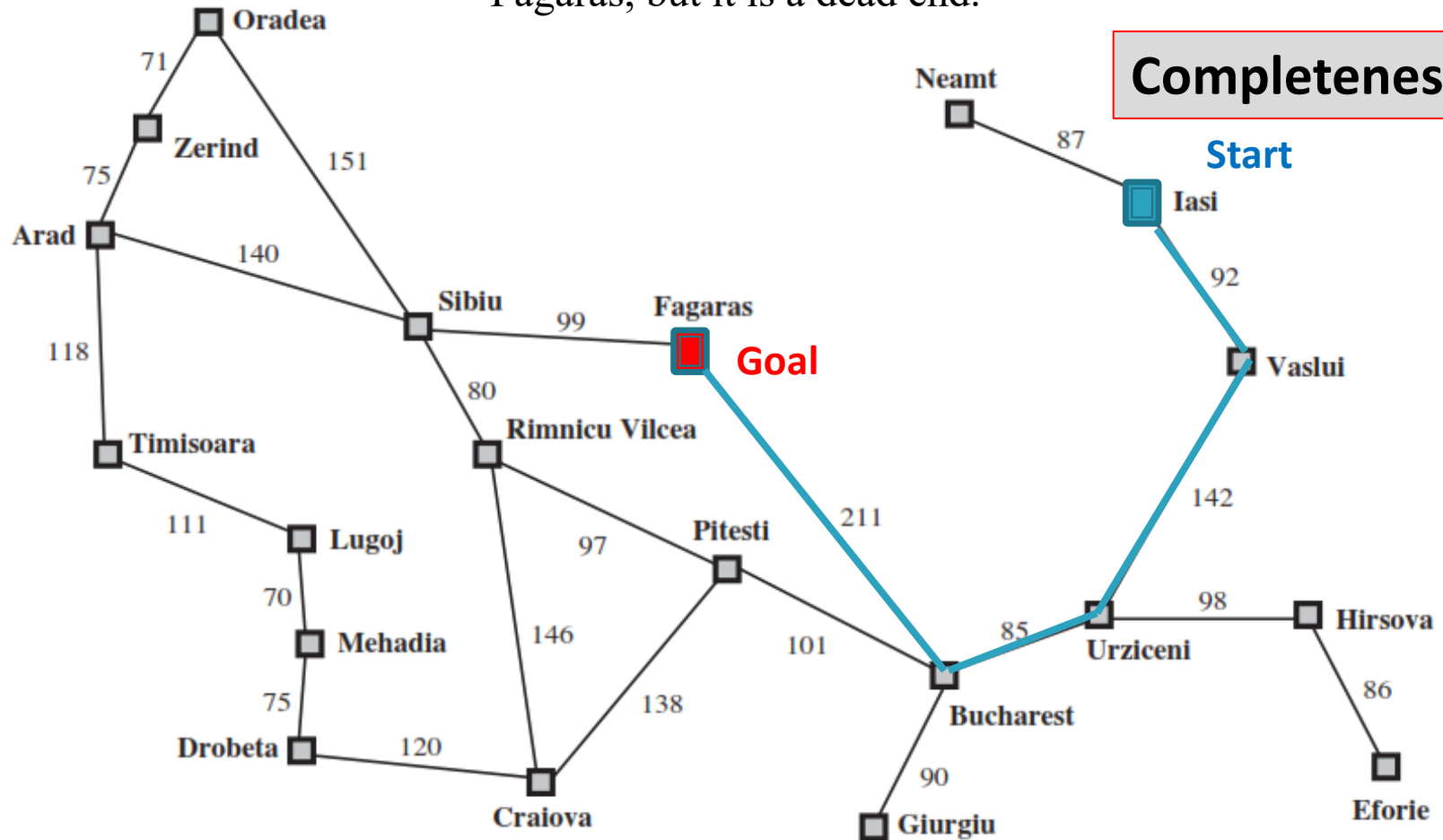


Completeness ?

A simplified road map of part of Romania.

Example 2

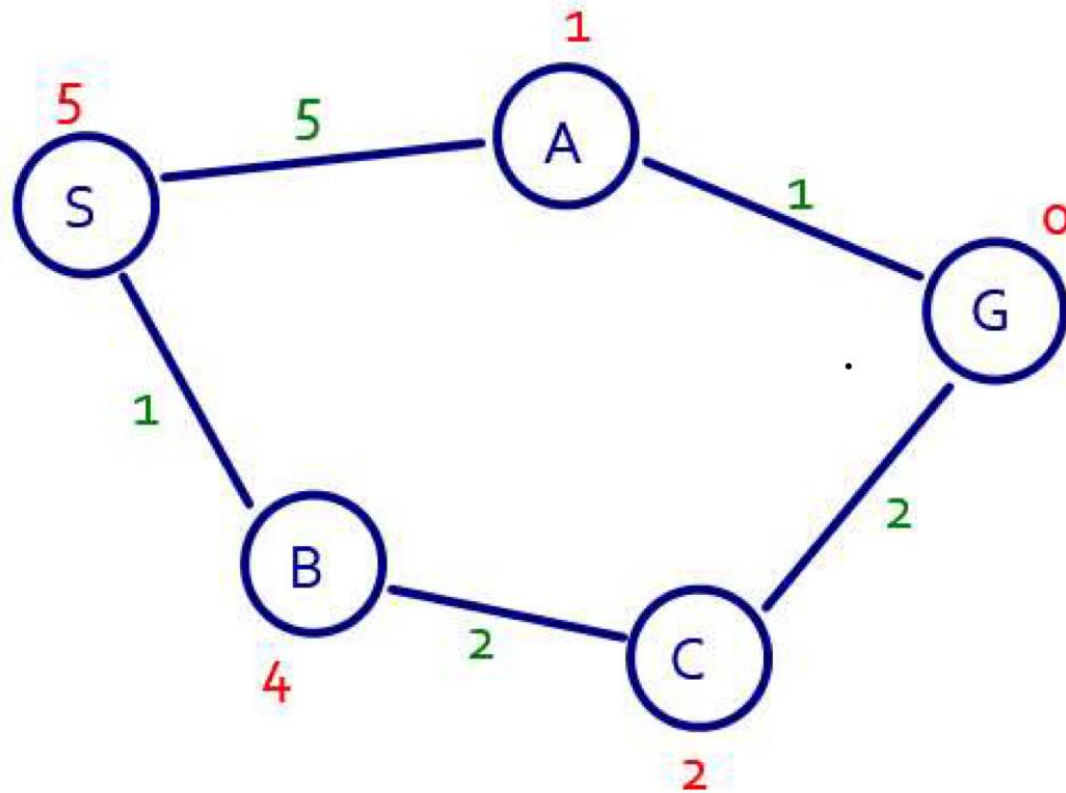
The heuristic--- **straight-line distances** ---suggests that Neamt be expanded first because it is closest to Fagaras, but it is a dead end.



A simplified road map of part of Romania.

Example 3

Optimality ?



A* Search

A* Search

- ▶ We **can bias Uniform-cost search** to find the shortest path to the goal.
- ▶ In fact, we are interested in by using a **heuristic function $h(n)$** which is an estimate of the distance from a state to the goal.
- ▶ It evaluates nodes by combining **$g(n)$** , the cost to reach the node, and **$h(n)$** , the cost to get from the node to the goal:

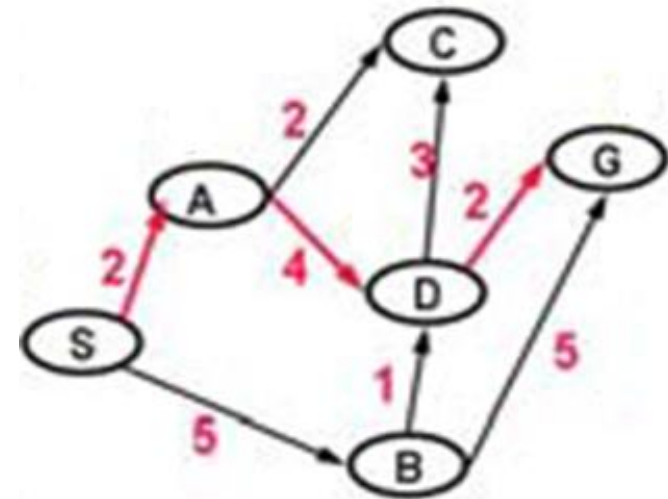
$$f(n) = g(n) + h(n)$$

A* Search

- ❑ Pick best (by **path length + heuristic value**) element of Q
- ❑ Add path extensions to Q

	Q
1	(<u>0 S</u>)
2	(<u>4 A S</u>) (8 B S)
3	(<u>5 C A S</u>) (7 D A S) (8 B S)
4	(7 D A S) (8 B S)
5	(<u>8 G D A S</u>) (10 C D A S) (8 B S)

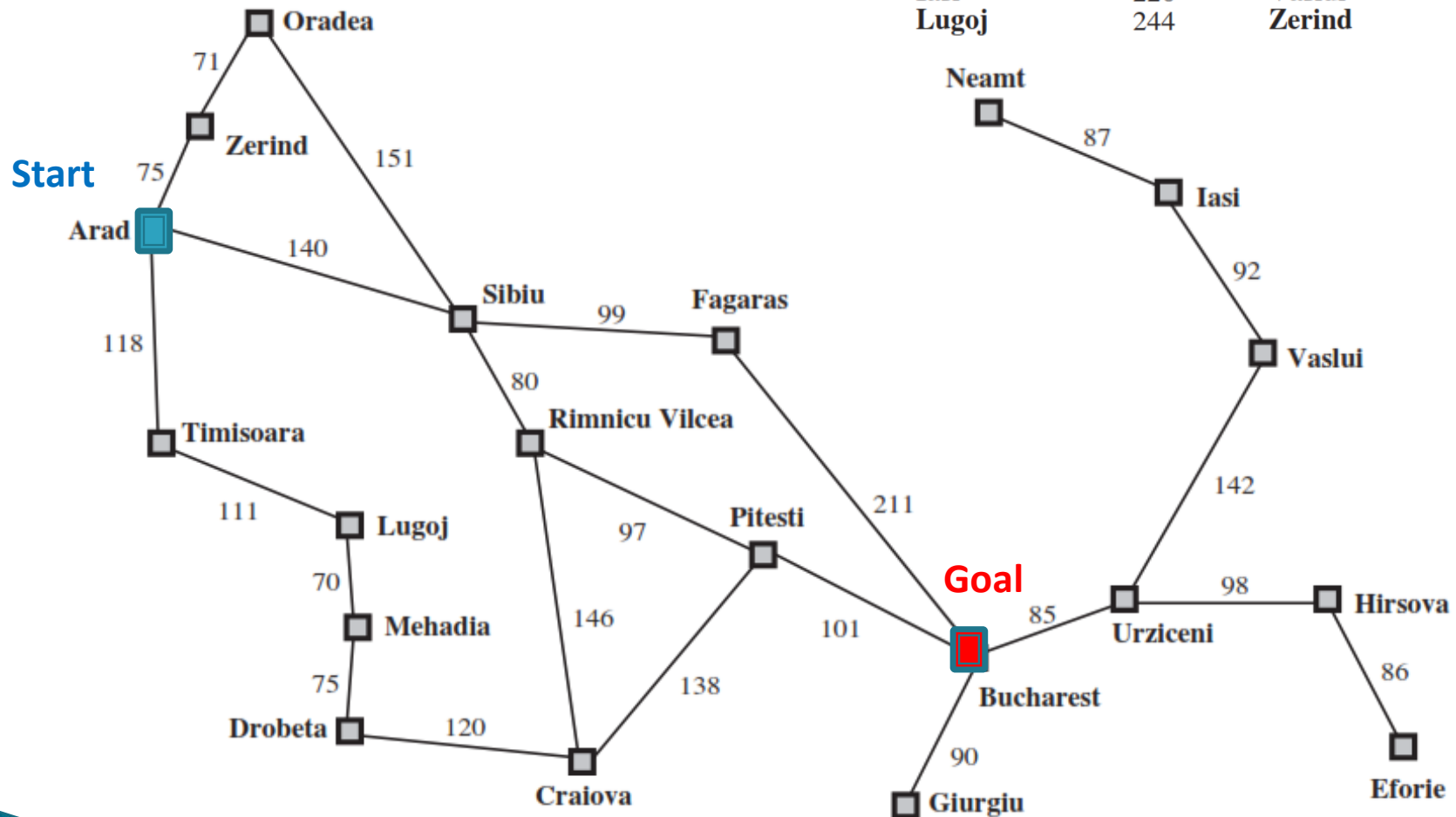
- ❑ **Blue Color** represents added paths
- ❑ Underline paths are selected for extension.



Heuristic Values

A=2 C=1 S=0
B=3 D=1 G=0

Example



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
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

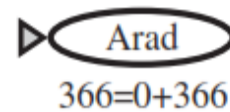
Example

Values of h_{SLD} —straight-line distances to Bucharest

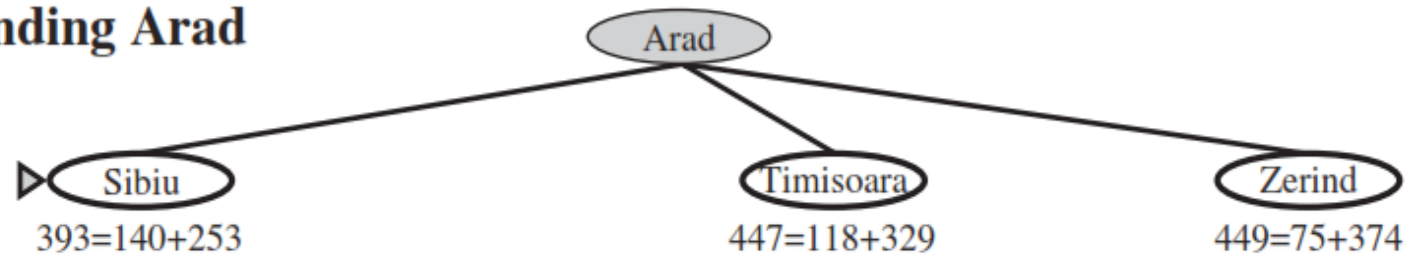
Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

Example

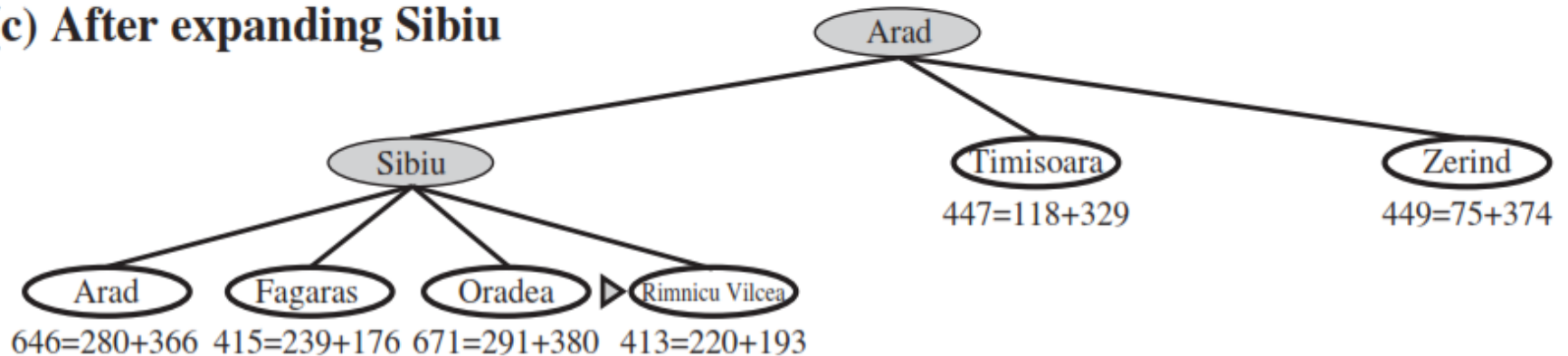
(a) The initial state



(b) After expanding Arad

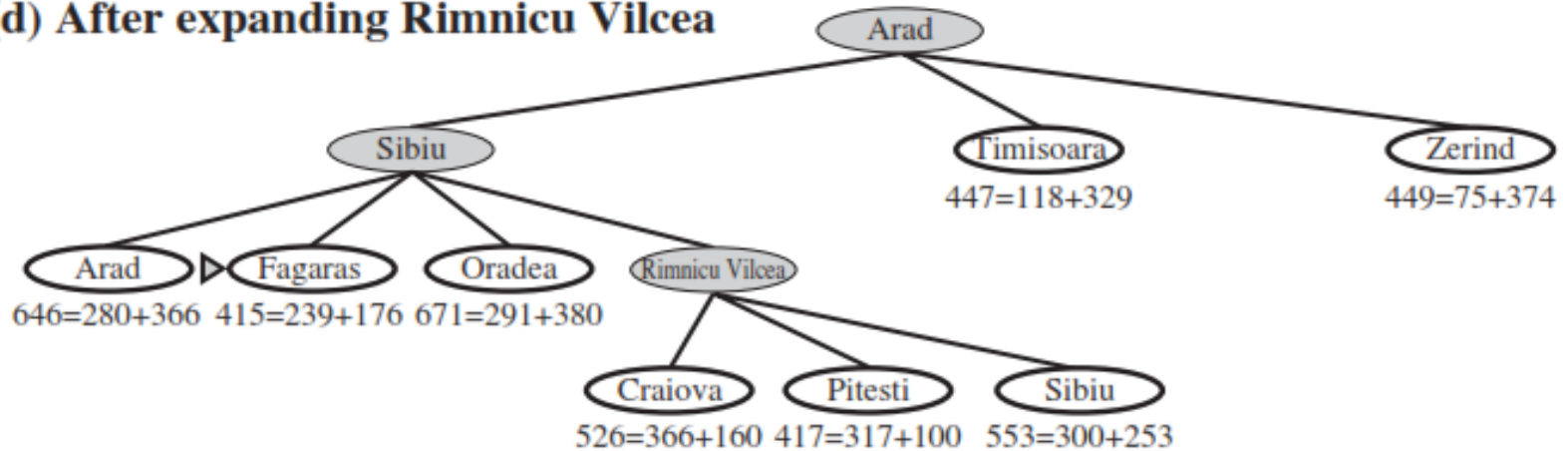


(c) After expanding Sibiu

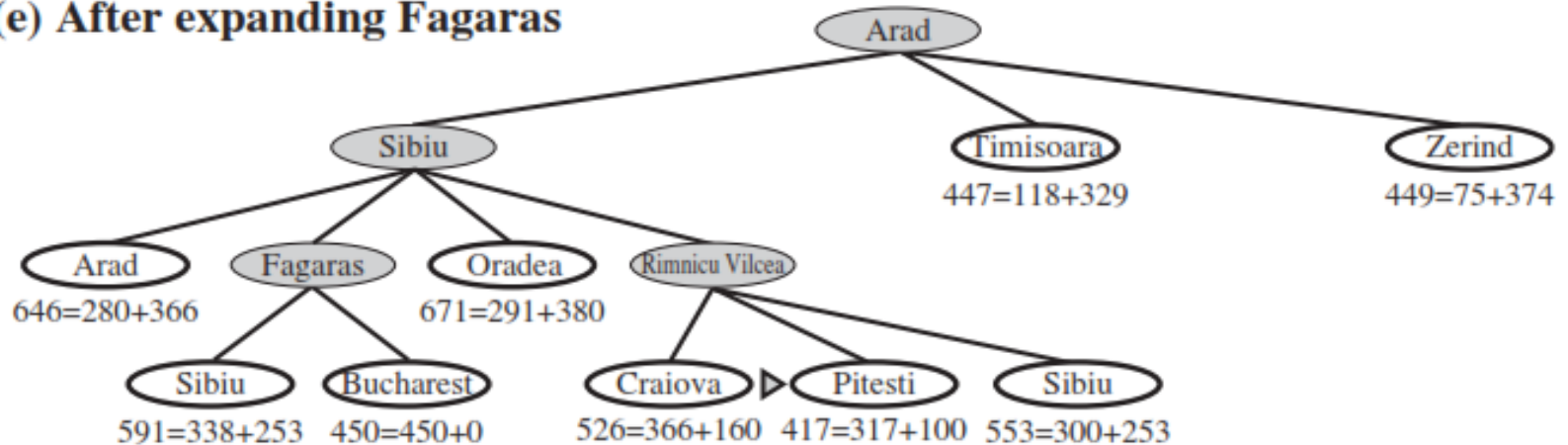


Example

(d) After expanding Rimnicu Vilcea

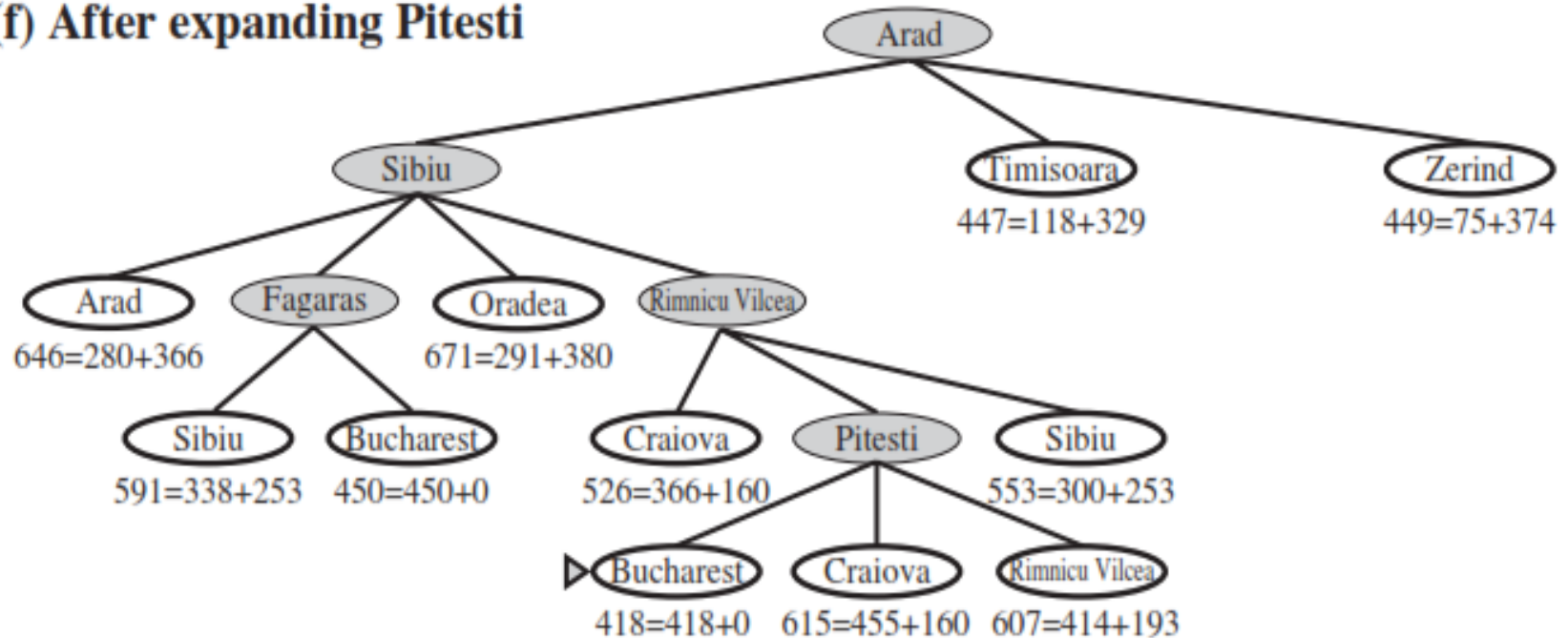


(e) After expanding Fagaras



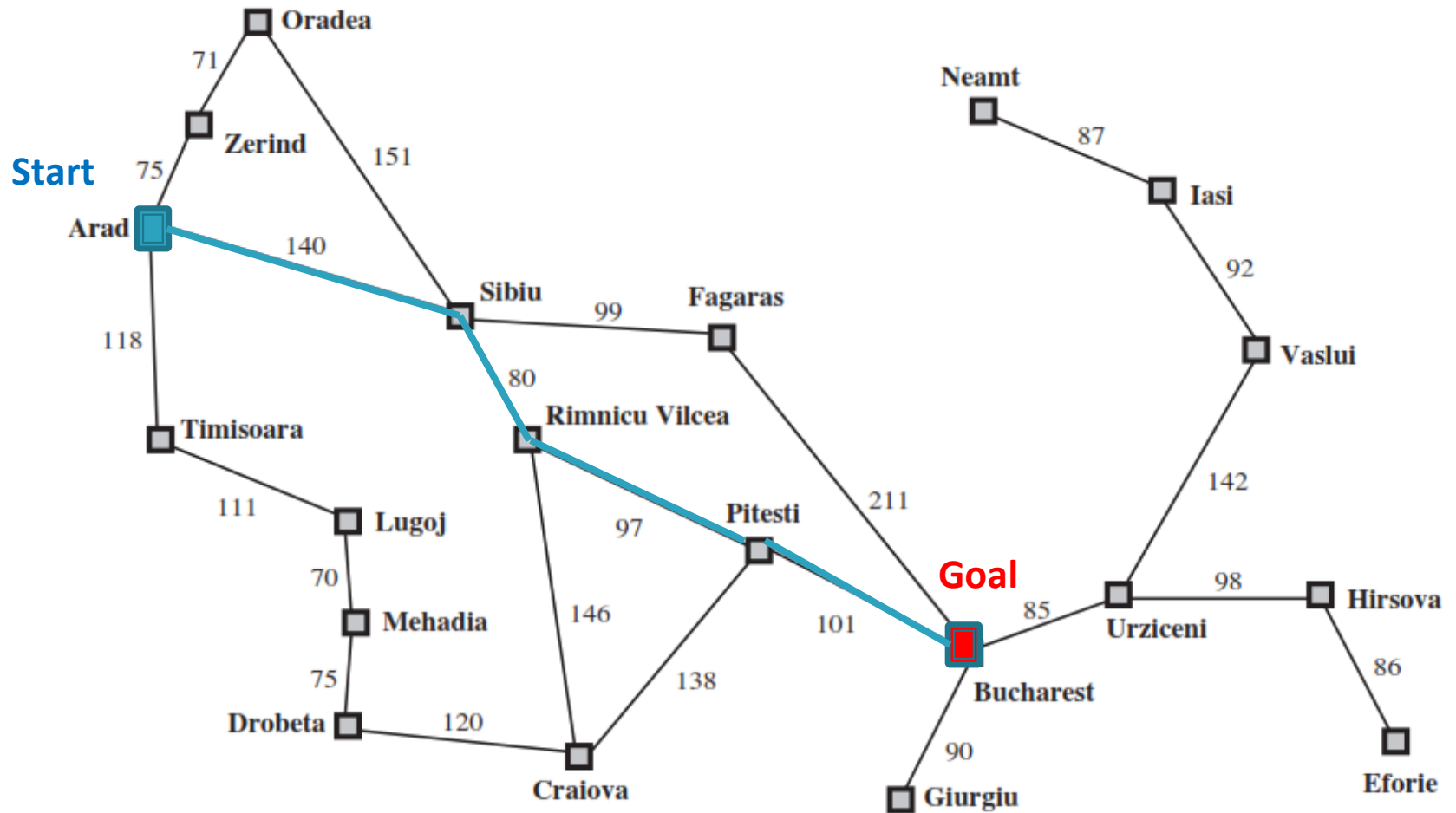
Example

(f) After expanding Pitesti



A* search for Bucharest: Nodes are labeled with $f = g + h$.
The h values are the straight-line distances to Bucharest.

Example



A simplified road map of part of Romania.

Reading Material

- ▶ **Artificial Intelligence, A Modern Approach**
Stuart J. Russell and Peter Norvig
 - Chapter 3.

